UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXV CICLO DEL DOTTORATO DI RICERCA IN

INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

PO FRIULI VENEZIA GIULIA - FONDO SOCIALE EUROPEO 2014/2020

Design and Simulation of Multilevel Interacting Mobile Robots for Space and Industrial Applications

Settore scientifico-disciplinare: **ING-IND/13**

DOTTORANDO
MATTEO CARUSO

COORDINATORE
PROF. ALBERTO TESSAROLO

SUPERVISORE DI TESI
PROF. STEFANO SERIANI

CO-SUPERVISORE DI TESI
PROF. PAOLO GALLINA

ANNO ACCADEMICO 2021/2022

Acknowledgments

I would like to express my sincerest gratitude to my supervisors at the University of Trieste, Dr. Stefano Seriani and Prof. Paolo Gallina, respectively, who guided and mentored me throughout my PhD program. Their door was always open whenever I had a doubt or problem and they were always ready to solve my doubts. Without them, their experience and knowledge, this work would not have been possible.

My sincerest gratitude also goes to the Regione Autonoma Friuli Venezia Giulia, which through the European Social Fund (ESF), financed me through a scholarship for the entire duration of the PhD course and therefore allowed me to continue my studies, which otherwise would not have been possible.

I would very much like to thank all the people and colleagues I met during my time abroad in Germany, at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), located in Munich. Among them, special thanks go to Dr. Ing. Martin Görner, who was my supervisor at the institute, for mentoring me, guiding me in my research, always being available and ready to solve any problem I had, and always making me feel at home.

I would also like to thank the DLR Entity for allowing me to use their instrumentation, infrastructure and resources to be able to pursue my research and experiments.

My greatest gratitude goes to my family, who always believed in me and stood by my side putting up with me, especially during the most difficult times. Thanks to their efforts, support, giving me strength and courage, it was possible for me to complete a long university journey, and I am sure they can be proud of that now.

I would like to thank Stefania, who was initially a friend and classmate and over time became a best friend. Thank you for the sincere friendship, and for always being a safe haven for me, always listening and advising, supporting and believing in me.

I thank my colleague, but most of all friend, Nicholas for sharing the office with me during the duration of my PhD, for the many laughs and best moments during this time, making the office a fun and pleasant place to work.

I would like to thank Luca for our friendship and for the unforgettable moments and laughter we had together throughout our time at the university.

I thank my friends Eleonora, Francesco, Matteo, Roberto, Nicola and Leonardo. Reaching this important milestone was easier also thanks to you. The moments spent together I will always carry with me.

I also thank all the friends from Trieste, the friends from Munich, the new

people who arrived, and also the people lost, who accompanied me and shared with me part of this journey, supported me and left me something beautiful to remember.

Contents

Abstract

When studying and planning to use mobile robots that need to be deployed in any application scenario, these robots, depending on the context, are always subjected to different types of interactions. These can be classified into three major groups: interactions between robots and their surroundings, interactions between the robot and other robotic devices, and interactions between robots and humans. Therefore, it becomes necessary, in the study and design phase, to take into account, study, model, and address these interactions in order to achieve proper and optimal performance of the robot, and to to avoid unpleasant situations.

In this dissertation, the interactions occurring on deployed mobile robots are addressed in different contexts. Specifically, this dissertation is divided into two main declinations: mobile robotics for space applications and for the industrial setting. Finally, with a focus on robot interactions, considerable attention is placed on the development of methods needed to address and solve them.

The main achievements obtained for mobile robotics for space applications field are the control of vehicles for space exploration in exotic applications, e.g. manipulation of objects via towing, and the study of the dynamics of complex robotic systems such as articulated planetary rovers and landers. On the other hand the main achievements for mobile robotics in industrial settings field are the development of a framework for the synthesis of a fleet manager that controls a fleet of tethered robots, and the development of a framework and pipeline for the development of smart controllers for mobile robots navigating moving crowds.

This dissertation is structured into four major chapters. The first provides an introduction to mobile robotics from a very general point of view, along with a classification of the various types of interactions that occur on robots. An overview of the use of mobile robots in the two macro fields of application considered, in general, is also provided, along with the problems and challenges characteristic of each of these fields. Finally, it provides a brief overview of the methodology and tools that will be used in subsequent chapters.

The second chapter focuses on the development and application of mobile robots in the context of space mobile robotics. The problem of soft-landing of an innovative robotic lander for space exploration is presented. Next, an innovative prototype named Archimede, of a four-wheel steerable rover is featured. Its design includes a suspension system, which ensures higher operational speeds. Moreover, every single aspect and component of the rover is first described, followed by a focus on steering capability and impact dissipation. The interaction between the wheels and the ground is then studied. Finally, it proposes a detailed study on

a technique for manipulating ground-resting loads, i.e. towing through elastic elements. In this context, the employment of this technique on the Archimede rover is studied and validated.

The third focuses exclusively on the development and application of mobile robots in the field of industrial mobile robotics. First, a problem characterized by robot-environment and robot-robot interactions will first be addressed. More precisely, a multi-robot problem is considered in which individual robots are chained together with cables, in order to meet the existing demands of a specific industrial context. Therefore, a high-level controller is presented, which assumes the role of the fleet-manager using techniques originally developed for redundant manipulators. This system aims to manage the fleet of mobile robots and at the same time allow them to perform tasks. Next, one of the many types of human-robot interaction is presented, i.e. the problem of navigating a mobile robot through a moving crowd. Specifically, the controller of the mobile robot is based on artificial neural networks, and trained in appropriate simulation environments using Reinforcement Learning techniques. The performance of the controller is then validated numerically and experimentally.

Finally, the fourth and final chapter will report the conclusions of this dissertation work and on the explored line of research.

Chapter 1

Introduction

1.1 Introduction to mobile robotics

Mobile robotics is a relatively new field and is a subgroup that belongs to the larger group of robotics. A mobile robot, as the name suggests, assumes the existence of a locomotion system integrated with it, which allows the robot to move and explore its surrounding environment, and in which it's designed to operate. These kinds of robots are more flexible compared to the classical industrial manipulators, which on the other hand are characterized by having a limited operating space, while that of mobile robots is potentially unlimited and they can perform their operations anywhere. However, since these robots must be able to move and interact with their surroundings in a safe manner they must be equipped with an appropriate perception system for sensing and understanding their surroundings, and they must also possess a certain degree of "intelligence".

Over the past decades, this area of robotics has seen a significant increase in interest from the scientific community, and subsequently, numerous studies have been carried out on mobile robots. This has resulted in considerable improvement of these systems, their inclusion in more and more applications, and to the development of more and more autonomous systems in the past decades. It is speculated that in the near future, these kinds of robotic systems may be extended further into every aspect of life, therefore helping humans perform their tasks and facilitate their everyday life.

Nowadays, there are different types of mobile robots, and an example of how they can be classified is based on the environment in which they are designed to operate [1]. Therefore, within the most common, there can be found ground-based mobile robots, also known as Unmanned Ground Vehicles (UGVs); aquatic or marine mobile robots, also known as Unmanned Underwater Vehicles (UUVs); and finally aerial or flying mobile robots, also known as Unmanned Aerial Vehicles (UAVs). Another environment in which they can operate is for space exploration purposes, and this field of robotics is usually referred to as space robotics.

Nowadays, ground-operating mobile robots constitute the majority of mobile robots, given their simplicity compared to other types of systems, and the greater number of available applications in which they can be put to. Despite this, lately the scientific community and others have been placing great attention on

(a) (b) (c)

Figure 1.1: In a) the QinetiQ-Talon-IV-EOD robot attempting on moving a simulated improvised explosive[1]; in b) the TRAMPER underwater vehicle (Wenzhöfer et al. [2]); in c) the MQ-1 Predator UAV[2]

UAVs and their potential applications, which will most likely see an explosion of their applications in the near future. The applications of ground-operating mobile robots, which are potentially unlimited, are numerous. Some examples can be distinguished: for example, one can find examples of domestic mobile robotics (see, for example, the rather common house cleaning robots), robotics for use in industrial environments (so for example mobile robots applied to logistics and transportation of materials for the production process), mobile robotics for applications in the health service (think of mobile robots that move along hospital corridors and transport medicines, or provide samples), in search and rescue applications, or in the military, and so on. Figure 1.1a shows a QinetiQ-Talon-IV-EOD (where EOD stands for Explosive Ordnance Disposal), as an example of mobile robot operating on ground. This is a tracked mobile robot used by the U.S. military that is being used for defusing operations.

UUVs, i.e. mobile robots which are operating in marine environments, instead, have been used for numerous applications, and in general, they are used in contexts that are considered dangerous or even unreachable for human beings, e.g. extreme marine depths. Additionally, some applications examples can be summarized in: survey operations, sampling for oceanic research, e.g. biodiversity, mapping of the seabed, climate research and monitoring, search for oil and gas resources, underwater pipeline inspection and their repair, wreck exploration, and so on. Moreover, these kinds of robots have applications in the military sector, for example, they can be used for ocean and underwater surveillance, placing or defusing underwater mines, anti-submarine warfare, and so on. As an example, Figure 1.1b illustrates the UUV TRAMPER which is a UUV autonomous benthic crawler equipped with oxygen sensors to perform long-term flux time series measurements at abyssal depth [2].

Finally, UAVs, or aerial mobile robots, are also nowadays used in multiple diverse applications. Some example scenarios that see their use are: transporting suspended payloads, performed alone or cooperatively with other UUVs; for aerial photography and film-making; search and rescue operations; archaeological

[1]DVIDS, Public Domain Dedication, via NARA & DVIDS PUBLIC DOMAIN ARCHIVE
[2]U.S. Air Force photo/Lt Col Leslie Pratt, Public domain, via Wikimedia Commons

sites exploration; public order surveillance; recreational use; delivery of goods and drugs; inspection of infrastructure such as buildings, pipelines, power lines; inspection of disaster sites; climate monitoring; agriculture and so on. As for the other two types of mobile robots, they are also widely used in the military, such as for surveillance and espionage operations and attack operations. A very famous example of this type of mobile robot UAV is the MQ-1 Predator UAV which is shown in Figure 1.1c.

Moreover, each of these three categories of mobile robots can be further classified into subcategories. Since this dissertation work is focused on ground and space mobile robotics, marine and aerial mobile robotics are left out because they are considered out of scope. For this reason from now on, only the mobile robots operating on the ground will be considered and described in detail.

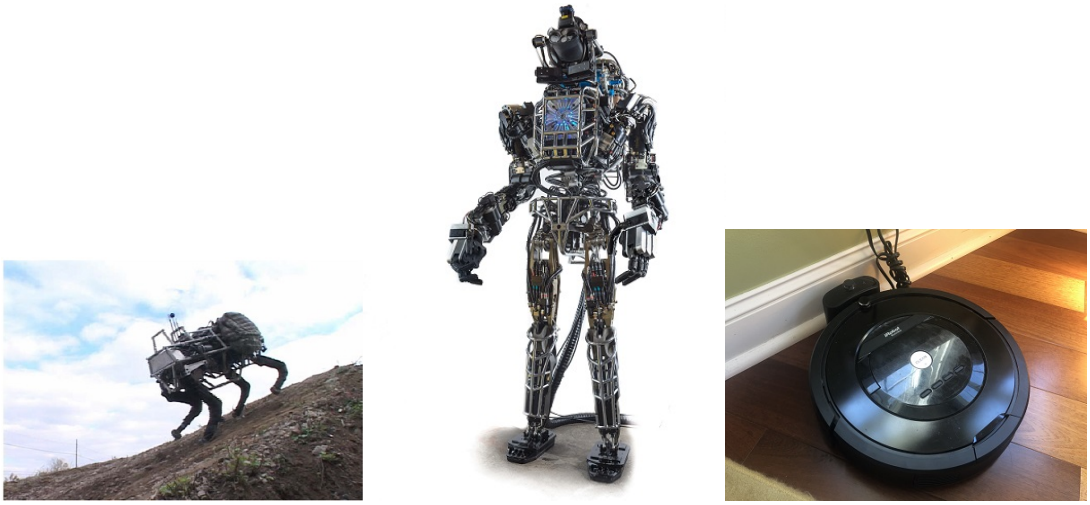
As for UGVs, these can be further classified depending on the type of locomotion system they utilize to move in the surrounding environment. According to Bruzzone and Quaqlia in [3], there are three main categories, which are:

- *Legged robots* (L): Legged robots are inspired by the movement of man or by the animal kingdom; moreover they use mechanical joints which guarantee their final movement. These kinds of robots have the advantage that they can be used vigorously on uneven and soft ground [4]. But at the same time they are a lot more complex as they possess many degrees of freedom and require a great deal amount of energy for their movement [5]. It follows that having a high number of degrees of freedom presents challenges also in control aspects. These kinds of robots can be further classified based on the number of legs that they use, e.g. one leg (hoppers), bipeds, tripeds, quadrupeds, and so on. In Figure 1.2a and Figure 1.2b are reported two examples of legged mobile robots. The first one is the quadruped BigDog [6], while the second is the well-known biped Atlas, both from Boston Dynamics. Another example is the quadruped Bert from DLR [7].

- *Wheeled robots* (W): Wheeled mobile robots (WMR) are characterized by the fact that they consist of a rigid body, usually called *chassis*, and by a system of wheels which fulfills the following two main tasks: providing the stability of the robot and allowing its movement in the surrounding environment. Wheels are cheap and simple and are very efficient on flat terrains. Moreover, from a control point of view, it is easier to control wheels rather than legged systems. When it comes to driving on uneven or soft terrain, on low friction surfaces, or in crossing obstacles other locomotion systems are preferred. Usually, they don't present balance problems, because usually these robots are designed such that the wheels of the robot are in contact with the ground at all times. Again these robots can be further classified based on how many wheels they use, and also on the type of wheels they use; which can be a standard wheel, castor wheel, Swedish or Mechanum wheel, ball, or spherical wheel. For a complete discussion about the wheels type and their characteristics, the reader is referred to [8]. In Figure 1.2c is reported the well-known Roomba wheeled mobile robot; which is a differential drive robot and is used in indoor environments for domestic cleaning

purposes.

- *Tracked robots* (T): Tracked robots, unlike those on wheels, use tracks to move in the surrounding environment. This type of locomotion is particularly suitable for motions on soft surfaces, uneven terrains, snow, sand, mud, etc. While they are not particularly suitable on flat and hard ground. In Figure 1.1a is reported a picture of the QinetiQ-Talon-IV-EOD, as an example of tracked mobile robot.



(a) (b) (c)

Figure 1.2: In a) the quadruped BigDog climbing a steep slope (Raibert et al. [6]); in b) the Atlas humanoid robot developed by Boston Dynamics and DARPA[3]; in c) the Roomba cleaning differential drive robot[4]

The locomotion systems described above are the main ones that can be found employed on common mobile robots, but those are not the only ones that have been studied and that can be found in the literature. More precisely hybrid systems can be found. The rationale for exploring and studying hybrid systems is to combine the advantages provided by both considered locomotion systems. These hybrid locomotion systems can be for example the ones used for Legged-Wheeled robots (LW) [9], Legged-Tracked robots (LT) [10], Wheeled-Tracked robots (WT) [11] and Legged-Wheeled-Tracked robots (LWT) [12]. For a full review about those systems, the reader is referred to [3].

Apart from the ones aforementioned, in the literature exist also other special locomotion systems, and all of them copy and take their cue from the counterpart biological system. For example, it is possible to find, snakes-like robots [13], worms-like robots [14], and so on.

Another way mobile robots can be classified is based on their degree of decision-making autonomy, the following are in fact distinguished:

[3]DARPA, Public domain, via Wikimedia Commons
[4]Piperpet, CC BY-SA 4.0, via Wikimedia Commons

- *Autonomous Mobile Robots* (AMRs): they have decision-making autonomy, and they do not need human intervention to perform the tasks they are required to do;
- *Non-Autonomous Mobile Robots*: differently from the previous ones, they do not have decision-making autonomy, and they require some level of interaction with the human. For example, Automated Guided Robots (AGRs) belong to this class.

In this section, a brief introductory overview of mobile robots, together with their classification, existing types as well as locomotion systems of UGVs have been provided. As this dissertation work is focused on terrestrial mobile robotics with locomotion systems that use wheels, therefore topics regarding other types of mobile robots will not be explored in greater detail. Moreover, this dissertation focuses attention on the application of WMRs in two main fields: space applications, and industrial applications. Hence, in the next sections a small introduction paragraph is reported about WMRs in these macro fields. Moreover, it will be provided a small section about the interactions that a mobile robot could be subjected to during its functioning.

1.2 Robot interactions

As already mentioned in the previous section, a mobile robot by its nature is designed to operate and move in the surrounding environment. For this reason, these kinds of robots are usually equipped with a large variety of sensors that are essential to them and allow the robot to "sense" and acquire knowledge of the surrounding environment. In addition, several different interactions occur throughout the life and functioning of the generic WMR, regardless of the environment and context in which they work. This section aims at providing a brief overview and description of the single kind of interaction that can occur. Furthermore, these interactions with the robot can be then classified basically:

- Environment-Robot Interaction;
- Robot-Robot Interaction;
- Human-Robot Interaction;

In the following the single interactions to which a mobile robot can be subjected will be described more in detail:

Environment-Robot Interaction

For what concerns Environment-Robot interactions, these can assume many different forms. Focusing the attention on WMRs, the most apparent and always present is the one with the terrain, through the contact processes between it and the robot's wheels. Undesired interactions with the environment may also happen, e.g. collisions with obstacles in the environment. Specifically, still regarding

the interaction with the terrain, in space application robotics impacts with the terrain can occur frequently and with different severity, together with normal obstacle negotiation operations. Moreover, in this contexts, the robotic system is frequently required to have other interactions with the environment such as performing drilling and rock and soil sample collecting. Another example of physical interaction of WMR with the environment occurs when performing manipulation tasks. If we think about a mobile manipulator, which has a robotic arm installed over the mobile base, it is possible to conclude that this robot is required to navigate to a target location, and once arrived it has to grasp and manipulate an object. In this way, the mobile robot is experiencing another environment-robot interaction through grasping. Some other operations that need interactions between the robot and the surroundings are: opening doors, assisted lifting of objects, and cleaning windows [15]. It is important to point out that the interactions with the environment are not exclusively physical, but can also take on other more abstract forms. For example, considering a robotic application inside a Smart Factory that highly exploits the Internet-Of-Things (IOT), the robot in this context may also interact with the environment through the network. For example, the robot can communicate its status and therefore trigger the loading/unloading process in a pallet moving application. In addition, the robot can receive and interpret signals from the environment such as the colors of the traffic lights in order to understand if it's free to move or has to stop and wait.

Robot-Robot Interactions

For what concerns Robot-Robot interactions, these can assume many different forms as well. These kinds of interactions arise when in the same operational space coexist two or more robotic systems. These interactions can be physical, desired, and undesired, or can be more abstract. As for the previous case, the most apparent physical and undesired interaction is the collision between two or more mobile robots that are sharing the same environment. In many contexts and for some applications multiple mobile robots can coexist in the same environment, forming a Multi-Robot System (in short MRS). In these contexts, sometimes, a given task can not be performed by a single mobile robot, but instead, a team of them is needed, such that they perform the assigned task in a cooperative way. Additionally, not only cooperation between mobile robots can be found, but applications can be found in which there are multiple robots performing their own operations, while at the same time being aware of the other robots. In these contexts, each robot (agent) must be capable of communicating and exchanging data with the other ones that share the same environment. In the literature, a large number of works can be found about cooperative MRS, and applied to every kind of application. These kinds of systems are really interesting because they can perform a task in a shorter time by exploiting parallelism, flexibility, and redundancy of the overall system. An example of this can be found in the research by Hichri et al. in which they presented a methodology to manipulate and transport by lifting payloads and exploiting cooperation between mobile robots [16].

Human-Robot interactions

Human-Robot Interactions (HRI for short) is a complete science in itself and is the last kind of interaction that a mobile robot may be subjected to during its operational life. HRI is a very hot topic nowadays and had been largely studied in the past years, and a lot of work about it can be found in the literature. This is because efforts are being made to develop and obtain robotic systems that are fully autonomous and as collaborative as possible with humans, and ultimately that do not cause harm and injury to them. Indeed flexible and dynamic settings, such as the ones of the Smart Factories, require the employment of several robotic systems that are considered "friendly" and can, other than coexist with persons in the same workspace, perform tasks in a collaborative way with them. Other than the industrial settings, another example of mobile robots which is being studied is to assist people with disabilities and to be employed in hospitals. In every setting that those systems will be deployed, HRI must be taken into consideration and addressed. An example in the field of HRI is given by the teleoperation of mobile robots. Specifically, approaches can be found that employ haptic interfaces for robot teleoperation [17, 18], eye-tracking for teleoperation without the need of hands and other input devices, but just with the eyes [19], speech signals recognition or augmented reality. HRI is a bi-directional type of interaction, and not only the human can communicate with the robot but the other way around happens as well. In fact, mobile robots can communicate their "intentions" in many ways to humans that are in their immediate vicinity. For example, they can use lights placed on the side of the robot and make them blink in the direction the robot wants to turn, exactly as with automobiles. Another example is that they can show by projecting on the floor the intended path they plan to follow [20]. The projected path can even be projected before it is actually happening, hence communicating to the humans their future states and giving them more time to react to the *signaled intention*. Other examples that can be found in literature, are leveraging the audio signals, the spoken language, the mounted displays, or the projection indicators [21]. Yet another example of mobile robot intent signaling, is provided by the research of Berg et al. Specifically, they have studied the problem of HRI, and proposed an approach in which the mobile robot receives input gestures and eye-tracking and outputs its information with a projector, in order to improve interactions with humans [22].

1.3 Fields of application

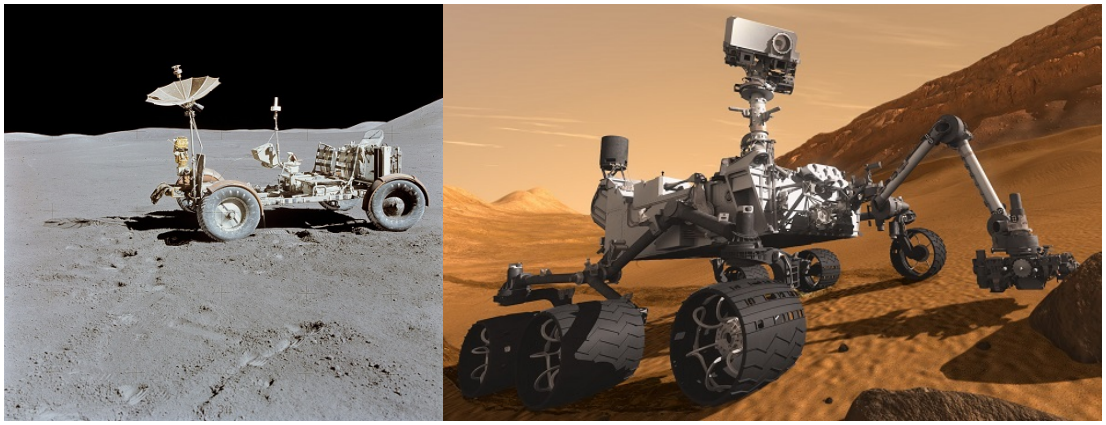
This section aims to summarize and describe the two main fields of application of mobile robots, which will be touched upon by the various topics throughout this dissertation. In fact, as will be seen in the following chapters different robots and different application field-specific topics will be explored separately. The two main fields of application of mobile robots considered in this dissertation are mobile robotics for space exploration applications, and mobile robotics in industrial settings, with a focus on the topic of *Smart Factories* and *Industry 4.0*. These two application fields will be discussed separately in the following paragraphs, in-

cluding for each a description of the environment, history, evolution, challenges, and objectives, as well as the interactions that the mobile robots can be subjected to in these settings.

1.3.1 Space exploration mobile robotics

The term *Space exploration mobile robotics* usually refers to that branch of mobile robotics that employs certain autonomous, semi-autonomous or telecontrolled vehicles used for planetary and celestial bodies exploration and generally during space missions, in which they are employed to complete a specific task in a given time window. These kinds of mobile robots are used to collect pictures, samples of the soil, dust and rocks, perform scientific investigations on the collected samples, and search for traces of water and life forms.

Two examples of real mission planetary rovers that have been used can be seen in Figure 1.3. More precisely, Figure 1.3a shows the Apollo Lunar Roving Vehicle which was used by astronauts in the missions Apollo 15, 16, and 17 during 1971 and 1972; while Figure 1.3b shows the Mars Exploration Rover (MER) Curiosity which has been used during NASA's Mars Science Laboratory (MSL) mission.



(a) (b)

Figure 1.3: In a) the Apollo Lunar Roving Vehicle[5]; in b) the Curiosity MER rover[6].

Rover requirements

Although mobile robotics for space applications or for planetary exploration, does not differ substantially from ordinary mobile robots operating on the ground, they are designed to operate in more hostile scenarios and are subjected to substantially more strict requirements. In fact, mobile robots for space applications need to face specific environmental issues resulting from: the need to operate in hostile and unstructured environments, such as micro-gravity and vacuum, loose soil,

[5]NASA/Dave Scott, Public domain, via Wikimedia Commons
[6]NASA, Public domain, via Wikimedia Commons

high level of dust, mostly unknown environments, large temperature changes; as well as technical issues related to mass, size and power restrictions, limited communication with the base and limited data storage, computational power and transmission capabilities as well as huge transmission delays, making real-time operations unfeasible [23]. Moreover, each component and the equipment must be designed to guarantee to be fault free for the whole duration of the mission. It follows that, besides the environmental constraints, these kinds of mobile robots are subjected to also mission-related constraints, such as lifetime and durability. In fact, a failure of parts of the rover is something highly unwanted, since from it can depend the success or the failure of the whole mission. Hence, as a consequence into a huge waste of materials, money, and resources. Therefore, it is required to these systems to be very reliable.

Localization and positioning

One of the key challenges in space application robotics is the accurate positioning of the mobile robot. Since global positioning approaches in these contexts are not possible or inaccurate, and then a planetary rover can not leverage the well-known GPS technology, they must rely on relative positioning approaches. In these systems, effective positioning is typically achieved by means of relative positioning such as *dead reckoning*, in which information coming from different sensors are fused together, typically by leveraging the Extended Kalman Filter (EKF), for better position estimation [24]. An example of this can be found in planetary rovers that use positioning systems integrating wheels encoders, or *wheels odometry* (WO), with Inertial Measurement Unit (IMU) gyroscopes and accelerometers readings [25]; or that integrates with sun sensors [26]. Although relative positioning can be achieved by just using the wheels odometry, it must be pointed out that this technique is characterized by having several problems and it's affected by errors. In fact, it is subjected to numerical drift due to integration errors, which accumulate; moreover, it is apparent that in this context this approach can be extremely compromised when the wheels undergo high wheel-slip conditions. The need follows to apply sensor fusion to integrate other sensors, in order to achieve accurate positioning. Moreover, positioning of the planetary rover can be effectively achieved also with *visual odometry* (VO) which is obtained by means of stereo-cameras [27]. Accurate pose estimation systems are essential for closing the navigation control loop of the vehicle [28]. Even though the majority of the positioning sub-systems are based on dead reckoning, some efforts in the past years have been done in the investigations of absolute positioning systems [29].

Navigation and path-planning

Another typical challenge that is related to planetary rovers is the one related to path-planning and navigation operations. As mentioned above due to communications constraints, i.e. transmission delay and bandwidth, these kinds of systems can not be constantly teleoperated remotely by a human operator. It follows that these kinds of systems must be able to guarantee a certain level of autonomy. As

a consequence of this, much attention has been posed by the scientific community and researchers toward the autonomy of these systems, although the development of fully autonomous planetary rovers is still an open challenge for robotics [30]. Specifically, current planetary rovers are not completely autonomous, but they are rather semi-autonomous systems. In fact, high-level path-planning operations and operations planning are conducted from the base station and then the best path is communicated to the rover. At this point, the rover has to move through those way-points. Usually, the autonomy of the rover is exploited in this operation, i.e. implements autonomous navigation through the commanded way-points.

Path-planning is a very well-known topic in the general mobile robotics field and is defined as the procedure to find an obstacle-free set of way-point that lead the robot from a starting point to a final one. Possibly this path is obtained as the best path which minimizes a certain criterion [31], for example, a traversability index, in the case of planetary rovers. Examples of available path planning algorithms for mobile robots are the A* [32], the D* [33], the RRT [34], *Potential Fields* based approach [35], *Probabilistic Roadmaps* [36], and approaches that use Digital Elevation Maps (DEM) [37].

Moreover, planetary rovers employ their own global path planners. They use sensors and cameras to generate a terrain map of their surrounding, then their path planner is responsible for finding an optimal traversable path which the rover must then follow [38].

Mobility and rough terrain negotiation

In the mobility challenge, understanding the wheel-soil interaction plays an important role, since it can affect the overall mobility of the rover, thus affecting the performance and the success of the planetary mission. Concepts of the classical terramechanics field [39], typically used for off-road vehicles, are usually applied to study the wheel-soil interactions of planetary rovers [40]. The correct understanding of the wheel-soil interaction helps also to characterize potentially harmful phenomena, such as wheel slippage. Specifically, many studies have been conducted through the past years in order to develop methods that allow us to estimate this phenomenon and compensate it, such as traction control [41], since due to excessive slippage the wheel can sink and lead the rover to end up stuck into the soil [42]. Additionally, within this field over the understanding of the wheel-soil interaction, it is required to properly design the overall locomotion system. Planetary rovers mostly use wheeled locomotion systems, rather than others such as legged or tracked ones, because they are simpler and have high energy efficiency, but are known to be characterized by low traversability [43]. Subsequently, a related challenge consists of the proper design of the suspension system, which can be passive, e.g. the well-known rocker bogie mechanism [44], or active [45]. It follows that passive suspension systems do not present problems related to the addition of dedicated actuators, additional control systems, and the necessity to reconfigure the structure, which after all introduces new points of failure into the system. When doing suspension system design, one tries to minimize mass, which as seen is a stringent requirement in this field, and at the same

time maximize performance, as well as ensure the highest level of traversability. In fact, the mobile robot is not only required to move over a horizontal granular terrain. In fact, it will need to be able to move in very unstructured environments, be able to overcome obstacles in its path without getting stuck in loose soil, climb up slopes, and move into craters. Consequently, the suspension system must provide these requirements, and at the same time ensure high stability of the robot, i.e., that it does not tip over. Other than the suspension system design it is needed to design the wheels of the rover. In fact, depending on the type of terrain on which the rover will have to operate, it will be necessary to design the diameter and width of the wheel, as well as the number, height, and shape of the grousers [46, 47], in order to maximize traction and minimize wheel's slip, hence the probability of sinkage, but also sustain loads and punctual forces coming from overcoming stones. Moreover, a distinction must be made between the design for only traversability purposes, and the design for also landing operations, thus requiring to resist to impacts, as was the case for NASA's MSL mission [48]. Finally, in this phase of the design process, must be chosen the actuators, the controllers of the drives, and related sensors.

Milestone missions

In the past years, many different planetary rovers have been used in real and successful space missions. In the following will be summarized the most remarkable ones. The Lunar Roving Vehicle (LRV), from the NASA agency, has been used for the first time in 1971 and is the first human-guided vehicle in space missions. Specifically, it has been used by astronauts during Apollo's missions 15, 16, and 17 respectively. Subsequently, in 1969, it was the turn for the lunar rover Lunokhod 1, successfully deployed on the moon, and passed to the history as the first-ever planetary rover remotely operated, that has been successfully deployed on a celestial body. After this remarkable mission, still the Soviet space agency successfully deployed in 1973 the second lunar rover, called Lunokhod 2. In 1997 the NASA space agency successfully landed and deployed the Sojourner rover on Mars, as part of the Path Finder mission [49]. Sojourner passed to the history as the first-ever rover that operated outside the Earth-Moon system. Still in the interest of exploring Mars, the NASA space agency, started the well-known Mars Exploration Rover mission, in which deployed on the Red Planet's surface two twin rovers in 2004, called Spirit and Opportunity respectively [50]. Later the same agency, as part of the Mars Science Laboratory mission, also landed in 2012 the well-known rover called Curiosity on Mars [51]. The last known and successful mission conducted by NASA is the Mars 2020 mission within which a rover called Perseverance and a drone called Ingenuity landed on Mars in 2021 [52]. Other remarkable missions that have been seen in the past come from the Chinese Space Agency. Specifically, within the space mission Chang'e 3 mission they successfully deployed the Yutu rover on the Moon's surface in 2013 and passed to the history as the first Chinese lunar rover. Subsequently, in 2019 they have landed the Yutu-2 rover, within the Chang'e 4 mission, on the far side of the Moon [53, 54]. Finally, the most recent mission is the Tianwen-1, in which they have successfully landed in 2021 the Zhurong rover on the Mars planet [55],

thus becoming the first Chinese rover to operate on another planet.

New concepts and research outlook

Apart from planetary rovers that have actually been used in real missions, many researchers have designed and studied mobile robot systems with potential use for the space exploration scenario. As an example, the Rocky-7 Mars rover prototype [56], the Micro 5 Lunar rover prototype, the DLR LRU rover [57], the DFKI's SherpaTT rover [58–60], or a reconfigurable robot for space applications [61]. For a full and comprehensive review of the past and present of planetary rovers, the reader is referred to [62].

Additionally, not only wheeled conventional wheeled planetary rovers have been investigated for space exploration applications. As for example, Massari et al. in their research have shown the development of a 6-legged mobile robot for space exploration [63]. Other examples include quadruped-legged robots [64], hybrid wheeled-legged rovers [58], or snake-like mobile robots for space exploration [65].

Within the field of *Mobile robots for space applications*, fall also different systems, such as the robotics landers. In fact, in some studies found in literature it has been considered highly versatile systems such as walking landers [66, 67], i.e. they work as a movable base.

One of the key challenges related to this applications field is the development of systems that grants faster mobility, which would grant a longer traveled distance, thus expanding the range of possible operations and experiments during a single mission [68]. Current planetary rovers are very slow, due to terrain constraints and energy and power requirements, but it is speculated that the development of faster planetary rovers will produce heavy repercussions on the research and development of systems for space exploration. A fast rover is referred to as one which can achieve stable speeds greater or equal to 1 m/s, which is ten to hundred times the operational speed of the current planetary rovers [69].

Another fascinating challenge, in the context of space exploration mobile robotics, is about the robotic planetary settlement setup which can be performed by only mobile robots, or by mobile robots and a human crew in a collaborative manner [70], also by using in-situ resources [71].

Summarizing the mobile robots employed in planetary exploration missions are subjected to mostly interactions with the environment. The predominant interaction is the one resulting from contacts between the vehicle and the soil, which occur in wheel-soil interaction for wheeled rovers or in leg-soil interaction for legged ones. Additional interactions with the environment occur when probing soil, rocks, and so on, e.g. with the embedded robotic arms; or when manipulation operations are carried out. Eventually, these mobile robots can be subjected to Robot-Robot interactions if employed in a Multi-Robot System, as is for example the case of the Perseverance rover and the Ingenuity drone. However, the contacts that occur between the soil and the vehicle are the most critical and need to be taken into account and carefully investigated. For this reason, the next section will provide a brief overview of the contact models between rigid bodies that will be used in the next chapters.

1.3.2 Mobile robotics for Industry 4.0 and Smart Factories

The general term *Industrial Mobile Robotics* refers to that branch of mobile robotics where mobile robots are studied and designed for use and can operate in industrial and manufacturing settings.

Typically and historically, the robotic systems installed in this field are of fixed type. Usually, they are robotic arms installed in assembly and production lines in contexts where there is a high repeatability of operations and the need to keep production volumes constant. Therefore, usually in factories, it is possible to find robotic arms which are confined in their working area, and it is possible to find them performing many different repetitive tasks such as: soldering, assembling, painting of parts, drilling, surface finishing, and so on.

Industrial setting and Industry 4.0

We are living in a time in which factories are in rapid evolution and are experiencing the fourth industrial revolution, which is commonly called *Industry 4.0*. This term was introduced for the very first time in Germany, specifically at the 2011 Hannover Messe [72]. The initiation of this process was made possible by the increasing digitization and level of automation achieved, the exponential increase in computing, and the ability to store huge masses of data. Moreover, it is also due to the fact that the global request has changed to a more customized production rather than a mass production. The term Industry 4.0 is based on nine fundamental pillars: big data and analytics, autonomous robots, simulation and digital twins, horizontal and vertical system integration, the industrial Internet of Things (IoT), cyber security and Cyber Physical Systems (CPS), the Cloud and Cloud Computing, additive manufacturing and augmented reality [73]. It is speculated that by the adoption of these technologies, the industries and manufacturing facilities that will accomplish the transition to Industry 4.0, will have enormous benefits in the future. These can be summarized in increased quality of the products, creating new business models, agility and flexibility, more efficient organization, manufacturing innovation, better and safer work environments, being more competitive in the global market, reduce costs, improve efficiency, reduce waste of resources, transition to mass customization from mass production [74, 75].

Smart Factories

The term Industry 4.0, comes also with a very important and well-known concept that is the one of the *Smart Factories*, referred to the manufacturing sector [76]. These are basically factories in which are implemented all the key features of Industry 4.0, with the objective of obtaining the aforementioned benefits. Within these contexts, the tendency is to have a production line that is highly automated, and exploit the IoT, i.e. where the systems, the sensors, and the machines are all interconnected through the internet, and they are allowed to interact and communicate with each other, and at the same time keeping a high level of flexibility

of the production lines and plant and of the processes, as well as customization. Additionally, IoT allows gathering huge amounts of data from every system in the network, which can therefore be post-processed by using Machine Learning (ML) and Artificial Intelligence (AI) techniques in order to optimize processes, production, and so on [77]. It follows that predictive models can be therefore be created, for example for maintenance purposes and machines auto-diagnostics, thus leading to a reduction of downtime, hence reducing costs [78]. Moreover, this can be exploited to perform continuous learning, thus having always better-performing systems.

Robotization and advanced automation

As mentioned above one of the pillars is the highly autonomous robotization within Smart Factories and production plants. It follows that a strong introduction of smart and autonomous robots, both fixed and mobile ones, into the industry has to be done. Specifically, mobile robots are gaining more a more popularity and are finding always more applications within this context over the past few years [79]. In fact, considering the requirement for flexibility in Smart Factories, the exponential evolution of technologies, increasingly sophisticated controls, automation, and the development of increasingly safe and autonomous mobile robots, has meant that these types of robots are gaining a foothold within industries precisely because of their flexibility. In fact, they have proved to increase manufacturer productivity and flexibility [80]. Nowadays, these systems are effectively employed and used for loading/unloading operations, material transportation and intralogistics [81].

Figure 1.4: Omron LD Series AMR[7]

Human-robot interaction

Another very important and key aspect that needs to be taken into account and faced when deploying mobile robots in industrial environments is the so-called "Human-Robot Interaction" (HRI), which has been introduced in the previous

[7]Auledas, CC BY-SA 4.0, via Wikimedia Commons

paragraph. In fact, these robots, do not always operate in a completely isolated environment, as perhaps might have been the case with the old manipulator robots that were typically fenced in. But in all likelihood, these will have to be employed in environments where there is not only the mobile robot, but other people who, along with the robot, are doing their work. This means that the robot should employ strategies and possess controllers to tackle effectively the HRI issue [22]. Moreover, the tasks assigned to the robot and the humans may be separate and independent, or it may be required that these two actors somehow share the environment and safely work together [82]. It is thus necessary for these robotic systems to be able to handle these interactions. Moreover, it is also likely that the robot presence will not be limited to one room, but it will need to be able to move between different areas of the industry. These considerations translate into the fact that the usage of mobile robots in the workplace shared with humans, leads to an increase in safety requirements [83].

Multi-robot environment and fleet management

Moreover, within a Smart Factory setting it's most likely that there won't be a single mobile robot, but rather a fleet of mobile robots [84], in which each mobile robot can either perform the assigned task on its own or it is required that they interact with the other actors of the fleet in order to perform the task in a co-operative way. The fleet is usually handled and controlled by the so-called fleet manager [84–86]. However, it follows that this context brings problems with the planning and management operations of the fleet. Moreover, since Smart Factories exploit the concept of the IoT, these mobile robots can interact, not only with humans or with their peers, but may interact with any smart machinery inside the network, e.g. trigger loading/unloading operations when the robot arrives at the station.

Outlook on Industry 4.0 robotics

In addition, several other more technical issues may arise which are not trivial to solve when a mobile robot is to be used, as for example: the safe navigation of the robot in an environment where other machinery, people, and other robots are present and can be highly dynamic; the continuous mapping of the environment; and most importantly, the continuous and precise localization of the robot in the environment.

Considering just the problem of robot navigation within these complex settings, it follows that the mobile robot must have a controller which grants it advanced obstacle avoidance capabilities, in order to avoid humans in the environment. It is apparent that the robot should always prefer to avoid crashing into humans and causing them injuries, rather than continuing the execution of the assigned task.

Figure 1.4 shows the Omron LD Series as an example of an Autonomous Mobile Robot (AMR) which can be used in industrial settings.

Summarizing mobile robots employed in these settings are subjected to a large variety of interactions: with other robots, with the environment in general, and

finally with humans. All these interactions need to be taken into account at a certain level and with the appropriate details.

1.4 Methodology

In this section is provided a brief introduction about the methodology and the tools that have been used throughout this dissertation. Specifically, the following topics will be touched within this section: the modeling of multi-body systems in which are presented the formulation for the dynamics of these systems as well as an introduction to two software packages that have been largely used in this dissertation; a brief overview on the soil contact models that are essential in the field of contact dynamics; the robotic middleware "Robot Operating System" (ROS) which has been used in every part of this dissertation; and a brief overview of the approaches that allow for the modeling of cables as elastic elements.

1.4.1 Modeling of Multi-Body Systems (MBS)

In robotics, and more in general in the field of mechanics, a general and complete functional system can be composed of multiple single bodies (which are often called *links*), which can be connected together through appropriate "connections" (which are often called *joints*), in order to provide them with relative motion. The overall system is then called a *Multi-Body System* (in short MBS), i.e. a system composed of multiple bodies possessing relative motion with each other. Given an MBS, in the 3D space, with n links and j joints, it follows that the "mobility" of a set of rigid bodies in the three-dimensional space, i.e. the number of DOFs for a kinematic chain is defined, by using the Chebychev–Grübler–Kutzbach criterion, as follows:

$$M = 6n - \sum_{i=1}^{j}(6 - f_i) \tag{1.1}$$

where, f_i is the degree of freedom of the joint i.

In literature can be found a lot of different formulations of the equations of motion for an MBS system. The most general formulation of the equations of motion for an MBS system, according to [87] is:

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{1.2}$$

where, $\boldsymbol{M}(\boldsymbol{q}, t) \in \mathbb{R}^{n \times n}$ is a positively defined matrix and is commonly referred to as the mass matrix, $\boldsymbol{q} \in \mathbb{R}^{n \times 1}$ is the vector of the n generalized coordinates, i.e. the number of *Degrees Of Freedom* (DOFs), thus $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ express its first and second order time derivatives, respectively; while $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ is referred to as the right-hand term of the differential equations and contains the external forces, as well as the Coriolis terms, t represents the time variable. The above equation results in a system of n differential equations of the second order.

The above formulation can be easily demonstrated for MBS forming an open chain (i.e. MBS systems exhibiting chains or tree topologies), however many MBS

are also composed of closed chains. The addition of these elements results in the introduction of additional constraint equations into the general formulation for MBS. Consider instead an MBS which contains closed loops. The formulation for the EOMs for these kinds of systems, in compact form, is defined as follow [88],

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{J}^T\boldsymbol{\lambda} = \boldsymbol{f} \tag{1.3}$$

where,

$$\boldsymbol{f} = \boldsymbol{f_{ext}} + \boldsymbol{f_c} \tag{1.4}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{m\times 1}$ is referred to as the vector of the Lagrange multipliers, $\boldsymbol{J} \in \mathbb{R}^{m\times n}$ is the constraint matrix, often called as the *Jacobian* matrix, $\boldsymbol{f_{ext}}$ is the vector of the external forces applied to the MBS, and $\boldsymbol{f_c}$ is the vector that contains the Coriolis and centrifugal forces and depends on the quadratic terms of the velocities. The term $\boldsymbol{J}^T\boldsymbol{\lambda}$ is explained as the vector of the constraint forces.

Since $\boldsymbol{\lambda}$ is a vector of the Lagrange multipliers and are unknown, then it is needed to introduce m more equations in order to solve the equations system in Eq. (1.3). Assuming a set of m independent algebraic constraints, these according to Nikravesh [89] can be expressed as:

$$\boldsymbol{\Phi}(\boldsymbol{q}, t) = \boldsymbol{0} \tag{1.5}$$

by differentiating Eq. (1.5) two times with respect to time is, therefore, possible to demonstrate the following relation:

$$\boldsymbol{J}\ddot{\boldsymbol{q}} = \boldsymbol{\gamma} \tag{1.6}$$

where,

$$\boldsymbol{J} = \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{q}} \tag{1.7}$$

where $\boldsymbol{\gamma}$ represent the right hand side of Eq. (1.6), resulting from the double time differentiation of Eq. (1.5). Furthermore, it follows that:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{J}^T \\ \boldsymbol{J} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_{ext}} + \boldsymbol{f_c} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{1.8}$$

The augmented system derived and shown in Eq. (1.8) is the system of equations that describe the dynamics of constrained MBS, which therefore needs to be solved and time-integrated in order to evaluate the motion of the MBS.

Simple MBS, with a low number of DOFs or particular systems that greatly simplify the systems of equations, can be solved analytically or simulated manually with a little effort from the user. However, when the number of DOFs increases, also the complexity of the system increases greatly and the derivation of the equations of motion becomes impossible, and such systems have to be simulated with appropriate software packages. Nowadays, exists a large number of software packages that are capable of simulating MBS, from the most simple to the most complex one, in an effective way. However, within this paragraph will be mentioned only two different software packages, since they have been used in the various parts of this dissertation. These are MSC ADAMS and Gazebo.

ADAMS

MSC ADAMS (Automated Dynamic Analysis of Mechanical Systems), owned by MSC Software Corporation, is a very common commercial software package and is well-known in the scientific community. It is largely used in the research community, but also highly used in industries. The solver runs in FORTRAN and C++. Its role is to solve the dynamics of complex MBS, and therefore the analysis of the distribution of the loads between the single parts. It follows also that it acts as a fundamental tool when designing a new mechanism, indeed it comes with utilities for parametric design and optimization.

ADAMS comes with a GUI, which drastically helps building the final system. In fact, within it, it's really easy to define new bodies, either rigid or flexible, specify their inertia parameters, connect them with many different joints, applying external forces as well as contact forces. All this is done in a visual way, thus extremely speeding up the process of building the MBS. Moreover, it offers the capability to import directly CAD models generated from other software packages, e.g. SolidWorks. It is apparent that this, from an end-user perspective, it's a huge key benefit.

ADAMS is largely used in the automotive field and in the field of vehicle dynamics, especially through its ADAMS/CAR, Adams/Chassis, Adams/Vehicle, and ADAMS/Tire modules [90–92]. Although this software package is widely leveraged in the automotive field for many purposes, e.g. optimizing suspension systems, due to its general-purpose nature it is also largely used in many other fields, for example, it has been used to simulate general mechanical systems such as a dynamic pointing system for satellites [93], to perform dynamical simulations on robotic systems, both industrial fixed manipulators [94] and mobile robots [95, 96], analysis of car suspensions [97], or perform dynamical simulation of soft-landing operations for planetary landers [98–102]. Another application in the field of robotics is provided by Bonilla et al. in 2014, which used the ADAMS software to simulate grasping with the Pisa/IIT SoftHand [103].

Moreover, ADAMS provides the user with a very useful and powerful tool that enables co-simulation with Matlab/Simulink. This is a key feature because it is possible to create the control algorithm in Simulink and then apply the control signals directly into ADAMS. An example of this can be found in the work of Song et al. in which the authors have used this feature in order to simulate a humanoid robot [104].

Last but not least, ADAMS doesn't come with just the Simulink co-simulation feature. In fact, for example, it provides also the ADAMS-EDEM co-simulation which permits the integration of ADAMS, which describes systems as MBS, with the capability to simulate and interact also with bulk materials simulated with DEM methods. This is very useful when it is desired to have a realistic behavior of the loose soil, for example for the optimization of heavy equipment, e.g. excavators, [105], in the field of the terramechanics in general or for investigations on vibratory fishing [106].

In Figure 1.5a can be seen a snapshot of a multi-body simulation done in ADAMS, for the planetary rover prototype Archimede, for the case of driving on uneven terrain and obstacle negotiation.

Gazebo

As stated at the beginning of this section, the other software package for MBS simulations that has been used within this dissertation is Gazebo which represents a really valid open-source alternative to the ADAMS software package described in the previous paragraph.

Specifically, Gazebo, which was first introduced by Koenig and Howard in 2004 [107], is an open-source multibody simulator, currently owned by Open Robotics, and it is mostly used to simulate robotic systems. Despite the large use of this software to simulate the dynamics of robotics systems, it can be used to simulate every kind of mechanical device, due to its multi-purpose nature, even in different environment elements. Additionally, the entire Gazebo dynamic simulator is written in C++, thus granting it to be very computationally high-performing. Moreover, a key benefit is that it allows the user to choose between different physics engines. Specifically, the ones included are ODE, Bullet, Simbody, and DART. This is particularly useful because, based on the problem it is desired to simulate, the right choice of the physics engine could lead to better and more accurate results.

Gazebo comes with a GUI, the *gzclient*, which can be used to visualize the ongoing simulation, interact with it, log data, and can be used to perform the modeling of the robotic system. Moreover, other than the model of the robot, it allows modeling of the environment, which is called the "world" in the Gazebo ecosystem. Inside the gazebo world, different objects can be placed and these can be either static, i.e. fixed with the environment, or dynamic, i.e. they can move because of the actuation or because of the interaction with another system. Moreover, it is possible to model all the desired kinds of environments, as an instance one can model the most simple environment (for example just a bunch of boxes placed here and there), passing to a very structured world (such as the exact recreation of an office or a building floor, comprising desks, doors, ornamental objects and so on), up to the most unstructured world (such as the recreation of a terrain from Digital Elevation Maps (DEMs), thus containing slopes, rocks, and so on). Although Gazebo allows the modeling of the robotic system directly inside its GUI, the best practice is to perform modeling by editing the SDF (Simulation Description File) files, which are basically XML files that describe the structure of the robotic system; the visual, inertial, as well as collision properties of the links of the MBS; the properties of the joints, i.e. ideal or not; and so on. Inside the SDF files the model of the robot is described by a tree or a chain topology, i.e. body1, joint1, body2, joint2, body3.

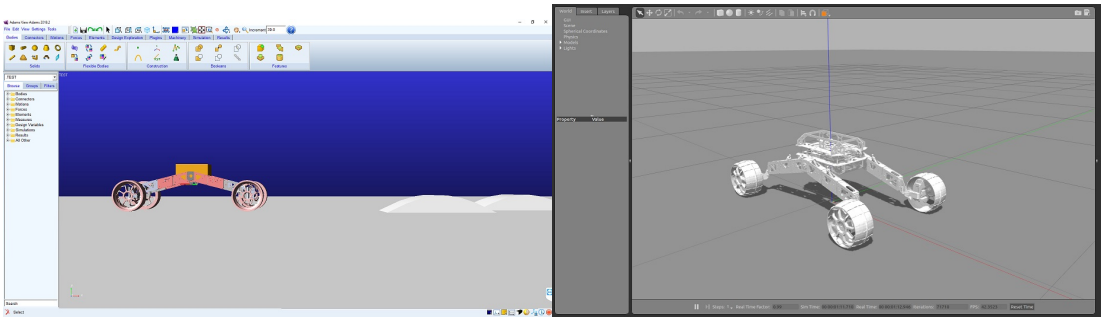
As already mentioned, this software package is very common and well-known in the robotic community and in research, and this is due to the fact that it is open-source and provides, apart from those described above, many features, such as the possibility to include inside the robot model multiple sensors, actuators as well as controllers, which mimics the real ones. Moreover, it allows the simultaneous simulation of multiple robots. Thus, multi-robot systems, fleets, and so on, sharing the same environment and interacting with each other, can be effectively simulated by utilizing this software package. As a consequence of these features, Gazebo is one of the natural choices among the various dynamic

simulators, for robotics. Moreover, Gazebo provides the possibility to extend the simulation environment and therefore add to it special functionalities which are obtainable by including opportunely user-written C++ plugins. Additionally, as a consequence of this, it can be easily coupled with other software packages. In fact, it is possible to find the Gazebo-Matlab/Simulink co-simulation feature, which is apparent to be very useful for control applications aspects, as well as the powerful ROS-Gazebo bridge, where ROS will be described in its dedicated section.

Additionally, the fact of being highly modular and potentially indefinitely extendable with the proper C++ plugins makes Gazebo a really versatile tool. In fact, Gazebo is not restricted to simulations of only robotic manipulators and wheeled ground vehicles, but with the use of proper plugins, it can be used to simulate systems operating in completely different environments: aerial, marine, or even micro-gravity. Specifically, for example, Manhães et al. developed a package called UUV Simulator, which extends the Gazebo simulator capabilities to the marine field and the simulation of UUVs [108]. In the same context Banu Theja et al., in 2022, analyzed the performance of an underwater vehicle by leveraging the integration of ROS and Gazebo [109]. Moreover, Zhou et al., in 2020, in their research proposed a framework for simulating UAVs which uses ROS and Gazebo simulation environment as well as an attitude controller trained with Deep Reinforcement Learning [110]. On the same field, Bernardeschi et al. in 2019 in their research studied the cooperation of UAVs by exploiting ROS and Gazebo simulator [111].

Other examples of the usage of Gazebo, as a dynamic simulator for robots, can be found: for the UUVs [108], for the UAVs [112–114], and an example of a simulation for a crawler [115].

In Figure 1.5b can be seen a snapshot of a multi-body simulation in Gazebo, for the planetary rover Archimede.



(a) (b)

Figure 1.5: Snapshots of the simulation environment for the Archimede rover: in ADAMS/View a); and in Gazebo b).

1.4.2 Soil Contact Models

In this paragraph is reported a brief overview and description of the two main methods typically used to model and solve the problem of the *Contact Mechanics*, as well as the main methods that will be used throughout this dissertation.

In the field of mechanics, the problem of contact arises when two distinct bodies come into contact with each other. As was mentioned earlier, in both fields of application of mobile robots that have been presented, i.e. industrial and space exploration, mobile robots are subject to this kind of contact phenomena, classified as interaction with the environment. While in mobile robotics for industrial applications contacts with the underlying surface is not so relevant, in the field of space exploration robotics, as mentioned, contacts with the soil play a fundamental role and need to be carefully addressed. In the latter case, for example for the case of a rover, the best practices in simulating these vehicles are usually the modeling of the rover as an MBS and the interaction with the soil with an appropriate Soil Contact Model (SCM) implementing the terramechanics theory [116]. However, for preliminary studies, and to study the overall behavior of the vehicle, such detailed and accurate models are not necessary, but models for rigid-body contact and MBS can be used.

It is apparent that the effects of contact interactions have a significant influence on the overall dynamics of an MBS, thus these phenomena and effects should be carefully modeled in order to obtain accurate simulations. Specifically, the motion of an MBS exposed to unilateral interactions, such as contacts and friction, is the subject of the field known as *contact dynamics*.

The contact dynamics is present in most practical examples and mechanisms. For example, it is present in wheel-soil interaction in the field of vehicle dynamics, robotic grippers, robotic hands, particles in Discrete Element Methods (DEM), legged and walking mobile robots, joint limits, internal contacts in the mechanism, and so on. As it can be seen contacts are not always unwanted phenomena but can be also functional, for example for the tasks of walking, grasping operations, or the motion of the well-known problem of the woodpecker toy.

The fields of "Contact Mechanics" and "Contact Dynamics" are science on their own, therefore a full and detailed review of these fields is out of the scope of this dissertation. On the other hand, a brief overview of the contact dynamics methods that will be used is provided in the following. In the literature can be found many approaches for solving contacts between different bodies. Those methods can be further classified into two main categories:

- *Non-Smooth Methods*: In these methods, the contact conditions are expressed as unilateral constraints. The resolution of impacts using these methods is usually done for example by solving a linear complementary problem (LCP). Contact condition can be for example taken into account by using the Signorini non-penetrability condition [117].

- *Penalty Methods (or Regularized approaches)*: These methods are based on the assumption that the colliding bodies are allowed to slightly penetrate one into the other, and a penalty is therefore applied to them in the opposite direction of the contact. These methods are simple methods, which

are based on the application of a reaction penalty force, and are computationally efficient. However, they come with some drawbacks which are that it is needed to tune the parameters which are not easy to estimate, and in some conditions, or with big step size can be numerically unstable.

It must be pointed out that within this dissertation, where it had been necessary to model and therefore solve the contact dynamics problem, only the penalty-based approaches have been used. For this reason, only these models will be considered from now on.

During the past decades, much attention has been posed by the scientific community in the field of "Contact Dynamics". Indeed, during this period a large amount of force-based contact models have been developed.

All the penalty approaches, i.e. the force-based contact models, found basis on the work done by Hertz on the contact mechanics. In the following just a few models will be outlined, while for a comprehensive and detailed review of all the contact models developed the reader is referred to the following research [118–120]. In his formulation, the normal contact force resulting from contacts between colliding bodies can be assumed to follow a non-linear elastic response [121].

$$F_n = k\delta^n \tag{1.9}$$

In Figure 1.6 is reported the schematic representation of the Hertz-Medelin frictional contact model between two colliding spheres.

Another very important formulation of the normal force is first given by Kelvin and Voight, and simply described by a parallel spring-damper system [122]:

$$F_n = k\delta + c\dot{\delta} \tag{1.10}$$

The same model can be further improved by considering it as a non-linear spring-damper system [122]:

$$F_n = k\delta^n + c\dot{\delta} \tag{1.11}$$

For what concerns the tangential friction force, this force usually is described with a model following the well-known Coulomb friction formulation, and it can be solved either with non-smooth approaches or with penalty methods. Moreover, during the past decades, many different formulations for the Coulomb-Friction model have been formulated, for example as step, ramp, or continuous functions. However, a full review of these models is out of the scope of this dissertation.

1.4.3 Robot Operating System (ROS)

This paragraph aims at providing a brief overview of the *Robot Operating System* (ROS), its features as well as its capabilities, since it has been widely utilized within this dissertation for the development of robotics applications, comprising mobile robot models and their control, and that are presented in the next chapters.

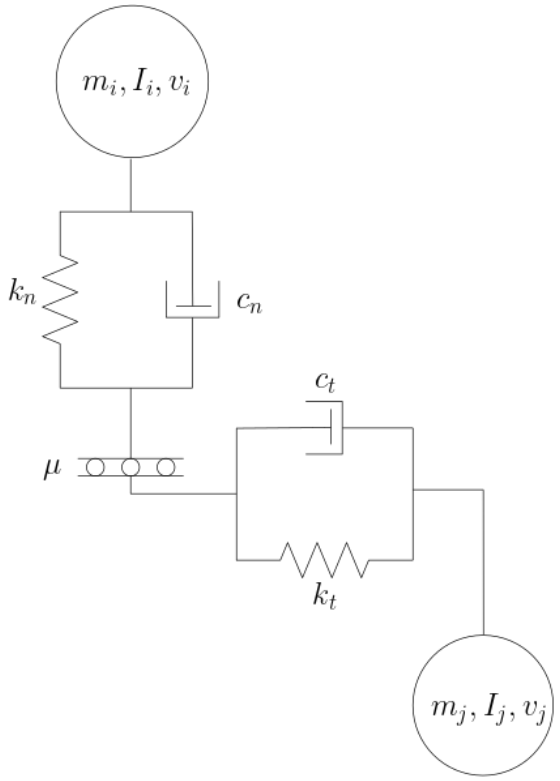


Figure 1.6: Graphical representation of the Hertz-Mindlin contact model between two colliding particles.

Robot Operating system has been first introduced by Quigley et al. in 2009 [123]. ROS is not an Operating system for robots, but it is a middleware or framework for the development of robotic applications and robot programming. Specifically, ROS comes with a huge set of libraries and tools, which help the final user build a robotic platform, which can be a prototype for research purposes or a commercially available platform. The tools and libraries available in ROS provide functionalities such as visualization, data logging, motion planning, drivers for the hardware such as sensors and actuators, communication, computer vision, machine learning, robot control and actuators control, simulation software, and perception. In Figure 1.7 is reported a schematic representation of the ROS ecosystem with emphasis on the set of tools it provides.

On the other hand, a very important feature of ROS is that it can also be seen as a communication layer between different processes. Moreover, it is flexible, and distributed and allows a robotic application to be highly modular. Specifically, every robotic application in ROS is scaled down in a set of separate nodes, where each node represents a single process that grants the correct functionality of a single component of the robot. In the ROS philosophy, each node should be independent from each other. Moreover, should be only responsible for a specific task. This fact allows the robotic application to be completely modular, hence the *crash*, i.e. the failure, of a single node in ROS will not compromise the execution of the whole system. As an example a node can be the process that deals with the calculation of the kinematics of the robot; or that deals with operating a

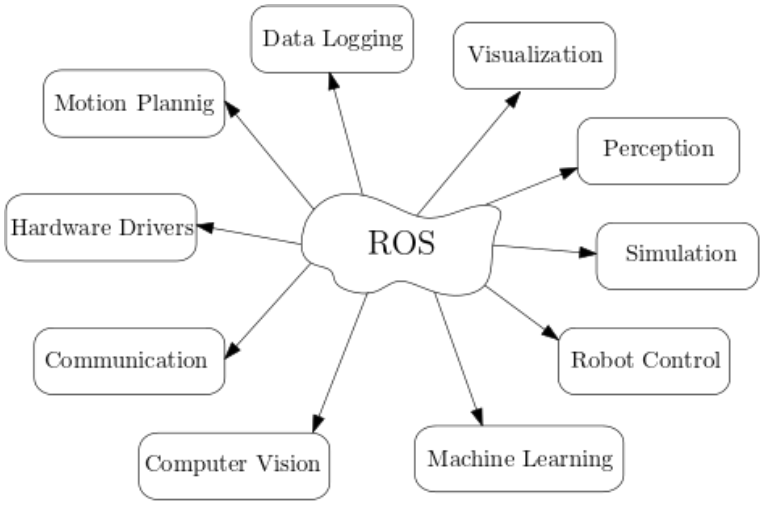


Figure 1.7: The ROS ecosystem showing the set of tools it provides.

sensor, extracting its readings, and processing the data acquired. Within a ROS application, each node is allowed then to communicate with each other by means of specific data structures, known as *messages*, exchanged through appropriate channels known as *topics* and *services*. Moreover, the nodes of a robotic system can also be distributed over a set of physically separated machines, and still, work and communicate with each other over the ROS network. It follows that thanks to these very important features (distribution and communication), ROS allows robot fleet applications to be developed quite comfortably. Another feature of ROS is that it is independent from the programming languages used. In fact, ROS applications can be developed by using C++, Python, Java, Matlab, and so on, leveraging the respective ROS clients. Another very important feature of ROS is that it allows being coupled with many other software packages through the appropriate ROS bridges. Specifically, it can be found ROS-Matlab, ROS-Gazebo, ROS-VREP bridges, and many others.

For a full and detailed discussion of the ROS framework topic, the reader is referred to the following [124, 125].

ROS is well known in the robotic community, in the research field, and in the industry. It is widely used in contexts such as academia and research institutes, due to its flexibility, modularity, for being open-source, and because of the large number of available packages that allow for fast robot programming and prototyping. On the other hand, some industries also have adopted ROS as part of the software for their robots. There is a large community, in which each individual can contribute with his projects and of course obtain modules and libraries needed for his applications. A large number of examples using ROS and robotic applications using ROS can be found in the literature. As for example Köseoğlu et al. in 2017 presented the design of an AMR whose software was based on ROS [126]. On the same line about the development of an AMR prototype leveraging ROS there is the research of Du et al. [127]. Another example of the design and development of an AMR based on ROS is provided by Kuzin and Sziebig in 2020, where their research presented SROS as a low-cost AMR for educational purposes [128]. An-

other claimed low-cost mobile robot based on ROS, called EUROPA is presented by Karalekas et al. in 2019 [129]. In the research from Mishra and Javed in 2018, the authors proposed a mobile robot, for service applications where the platform is required to map the structured surrounding environment, localize itself and then navigate in it [130]. Su et al. in 2021 presented a simulation environment based on ROS, Gazebo and RViz [131].

Other applications example is provided by Beraldo et al. in 2018, which in their research presented ROS-Health as a framework for neurorobotics, build over the framework ROS [132]. Marian et al. in 2020 used ROS to control a mobile robot in the Gazebo simulation environment [133]. Yao et al. in 2015 used ROS in order to test multi mobile robots collaboration algorithms and therefore simulate a game of robot soccer in the Gazebo simulation environment [134].

1.4.4 Cable modeling

This paragraph aims at providing the reader with a brief overview of some methods that can be used to model flexible elements like cables, and which are used within this dissertation. Cables are interesting elements, which possess a great level of flexibility. It follows that due to their great flexibility they can not be treated as rigid bodies, but other kinds of approaches must be used. In literature can be found a large number of works presented by many researchers which try to model kinematically and dynamically the cables. The models available in the literature differ from each other by the complexity which is introduced in the cable model, as instance one can take into account or not also torsional and flexural stiffness of the cable, and the assumptions made. Within this paragraph are reported three different methods among all the existing ones, which have also been used in the next chapters of this dissertation. These models are respectively the catenary model, the lumped-mass model, and the model as an unilateral constraint.

- **Cable as a catenary curve**: This method for cable modeling is based on the use of a mathematical relation to describe the shape of the cable in static conditions. This mathematical relation describes a curve which is called *catenary*. As already stated this approach defines only the shape of the cable and allows the extraction of the tensions of the cable in equilibrium conditions. This approach cannot be used to describe very well the dynamics of the cable. Although many different and extended formulations of this cable model have been done, e.g. elastic catenary models [135], in the bi-dimensional case the most general equation for the catenary curve is defined as follows,

$$y(x) = a \cosh\left(\frac{x}{a}\right) \tag{1.12}$$

  where y is the ordinate, x is the abscissa of the catenary curve and a is a coefficient, which is defined as:

$$a = \frac{T_0}{\lambda g} \tag{1.13}$$

where T_0 represent the horizontal tension of the cable, λ its specific mass and g the gravitational acceleration. Although, the great simplicity of this cable model, it has been largely used by different authors for the applications in their research. An example of this can be found in the work from Su et al., which applied catenaries to a suspended parallel camera robot system [136]; or in the work from D'Antonio et al., in which they instead considered catenaries between two quadrotors [137].

- **Discrete lumped spring-damper-mass approach**: In this approach the cable is divided into n particles having mass m_i. Each of these particles is then connected to the previous one and the next one through a parallel spring-damper system, responsible to keep the particles close to each other. It is worth noting that by using this method it is possible to model in a very easy way a flexible tether. Moreover, it is possible to model its dynamics and derive its equations of motion. The system of EOMs can be expressed in a matrix form as follows,

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{F} \tag{1.14}$$

  where $\boldsymbol{M}$ is referred to as the mass matrix, $\boldsymbol{C}$ is referred to as the damping matrix, $\boldsymbol{K}$ is referred to as the stiffness matrix, $\boldsymbol{x}$ is the $3n$ vector of the generalized coordinates in the Cartesian space, and finally, $\boldsymbol{F}$ is the vector of the external applied forces.

  Another advantage of using this method is that it is possible to include the already-seen force-based contact models, to simulate scenarios in which the cable touches the ground or hit some obstacle. In fact, the contact models are included in the dynamics of the system by just computing the contact forces and then applying them to the system as a set of external forces directly to the interested particles. As an example this can be seen in Figure 1.8, where a cable suspended between two different height hanging points has been discretized in 20 elements and left free to lye on the floor.

  This approach has been largely used in research studies involving mooring cable systems [138–140]. For example in this field of application hydrodynamical forces, together with contact forces and gravitational forces, take place on the cable nodes, and influence its dynamics. For a comprehensive discussion about the lumped spring-damper-mass model for the dynamics of an elastic cable, the reader is referred to the following review [141]. Another approach that must be reported, and very similar to the just described lumped-damper-spring model, is the one in which each body of the cable is connected to the other by means of a rigid link. An example of this approach can be found in the work of Goodarzi et al. in 2015, in which they studied the problem of transporting a payload with a multicopter and a cable from a control perspective [142].

- **Unilateral elastic constraint (traction only condition)**: This is the most simple model which can be used to describe an elastic cable element.

The idea behind this model is that since the cables are flexible elements, which can exert tension only when pulled, but when compressed they do not exchange forces, but go slack, it is reasonable to model the cable as an unilateral constraint. This formulation considers an elastic cable that is in continuous traction condition, i.e. the cable is always taut. Consider the two anchor points of the cable as $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$, its length at rest is defined as l_0, while the equivalent stiffness of the cable is $k_{c,eq}$. The model is then defined as follows,

$$\boldsymbol{T} = \begin{cases} k_{c,eq}\left(l_0 - ||\boldsymbol{p_i} - \boldsymbol{p_j}||\right)\frac{\boldsymbol{p_i}-\boldsymbol{p_j}}{||\boldsymbol{p_i}-\boldsymbol{p_j}||}, & \text{if} \quad ||\boldsymbol{p_i} - \boldsymbol{p_j}|| \geq l_0 \\ \boldsymbol{0_v}, & \text{if} \quad ||\boldsymbol{p_i} - \boldsymbol{p_j}|| < l_0 \end{cases} \tag{1.15}$$

where $\boldsymbol{T}$ is the tension vector exerted by the elastic cable. The model described by Eq. (1.15) basically states that when the cable is taut, it experiences an elongation, hence it exerts a tension force that follows an elastic law of the form $T = k_{c,eq}\epsilon$. On the other hand, when the cable goes slack, then the tension force is considered to be null, disregarding the real behavior of the cable and its shape. It can be stated that since this model is elastic it follows the Hooke's law, hence the equivalent stiffness is defined as follows,

$$k_{c,eq} = \frac{EA_c}{l_0} \tag{1.16}$$

where A_c is the cross-section of the cable and E is the Young modulus of the elastic cable. Although this model of the cable is really simple and it introduces many simplifications and assumptions, it has been used in several works. An example of this can be found in the research from Lin et al., in 2020, in which they used this formulation to model the cables of a Cable-Driven Parallel Robot (CDPR), used for throwing operations [143].

1.5 Outline

Within this section will be provided a brief outline of how this dissertation is structured, as well as the studies and topics that will be covered in the next chapters. Moreover, it will be provided also for each of the topics that will be covered a small summary about their novelties and contributions to the state of the art.

Specifically, in this dissertation, the attention is posed to the different interactions that a wheeled mobile robot is subjected to or may be subjected to during its operational life, and about their effects on it and try to address them. As it has been said in the earlier sections these interactions can be classified into robot-environment interactions, robot-robot interactions, and human-robot interactions. More precisely, all these three types of interaction are, in the following chapters, considered in specific and different applications and therefore conducted studies intended to solve problems of scientific interest. In addition, each of the

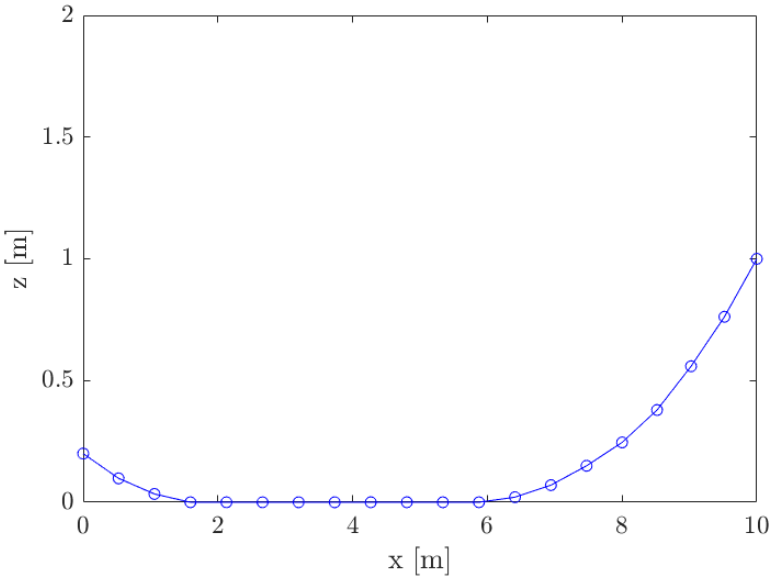


Figure 1.8: Lumped mass model for the cable, laying on the floor.

interactions seen is applied in both presented contexts of mobile robotics applications, namely, mobile robotics for space applications and mobile robotics for industrial applications.

More specifically, this dissertation is subsequently divided into two large parts, namely Part I and Part II. The first part introduces and presents three different studies and applications of mobile robots for applications in space missions and planetary exploration contexts. The second part, on the other hand, introduces and describes two different studies and applications of mobile robots for applications in industrial contexts.

Going into more and more detail, Part I consists of several sections, each describing and addressing a specific problem of mobile robotics applied in space exploration contexts and focusing on the specific interaction the mobile robot is subjected to. These sections can therefore be listed as follows:

A. *VRD-based lander*: In Sec. 2.2 is firstly presented the design of a robotic lander whose soft-landing mechanism leverages the *Variable Radius Drums.*. This robotic system under investigation is subjected to Robot-Environment interactions, due to contact with the soil. Subsequently, it will be developed an analytical-based dynamics model, along with a description of the modeling of the lander in ADAMS. The proposed soft-landing mechanism will be then evaluated for two different test cases, and the analytically-based model will be numerically validated through direct comparison with the ADAMS one.

B. *Archimede planetary rover*: In Sec. 2.3 is firstly presented the design of a four steerable wheels planetary rover, called Archimede, whose suspension system is based on the concept of the S-Structures. Along with the design is described in detail also the hardware and the software parts of the mobile robot, which is completely developed in ROS. Moreover, it will be described the odometry subsystem, and the control approach of the mobile robot that is

based on the Ackermann steering and the ICR projection technique to fulfill the joint's limits avoidance additional task. Subsequently, it will be developed an analytical-based dynamics model, along with a description of the modeling of the rover conducted in ADAMS and Gazebo software packages. The odometry subsystem will be then validated numerically first, as well as in Gazebo dynamic simulations, therefore it will be validated experimentally with drive tests in a sandbox. Finally, the analytical-based and the ADAMS numerical dynamic models will be validated through outcomes comparison with the experimental prototype, for the cases of two impact tests, and obstacle negotiation. Finally, this robotic system under investigation is subjected to Environment-Robot interactions, due to continuous contact with the soil.

C. *Towing Manipulation strategy*: In Sec. 2.4 will be presented a strategy for objects manipulation through towing operations by means of mobile robots. A formal modeling of this problem will be first presented leveraging the quasi-static motion assumption. Therefore, two dynamics models will be developed. As a case study, it will be considered the Archimede rover as a test bed for towing the objects. Therefore controllers for trajectory tracking based on the ICR projection will be described and introduced in the control loop. Subsequently, the dynamics model's outcomes will be compared to quantify the deviation from the quasi-static approach. Finally, the presented approach is validated experimentally. Within this application, the robotic system is subjected again to Environment-Robot interactions, due to: on one side to contact with the soil, on the other side due to interactions with the towed object through the towing medium.

On the other hand, Part II consists of two sections, each describing and addressing a specific problem of mobile robotics applied in industrial settings and therefore focusing on the specific interaction the mobile robots, that have been considered, are subjected to. These sections can be listed as follows:

D. *Tethered Mobile Robots*: In Sec. 3.2 will be firstly presented a framework, and a detailed and formal formulation, that leverages an analogy between the Tethered Mobile Robots system and a redundant manipulator, and allows for the creation of a fleet manager. The control approach will be presented, including the constraints as additional tasks. Subsequently, based on an industrial request, a case study will be selected and the presented approach validated over four different scenarios. Finally, the approach is validated through dynamic simulations in Gazebo dynamic simulations. Within this robotic application the mobile robots are subjected to two kinds of interactions: firstly the Environment-Robot interaction due to contact with soil, and proximity to obstacles in the environment; secondly to Robot-Robot interaction through the tether connection and the proximity constraint.

E. *Crowd Navigation*: In Sec. 3.3 will be firstly described and modeled, in a simulated environment, the problem of a mobile robot navigating a crowded environment with moving pedestrians. Subsequently, the approach chosen,

based on reinforcement learning, is described together with the choice of the used neural network. Subsequently, the neural network training and testing results are shown, validating numerically the presented approach. Finally, the approach is validated experimentally in a real setting, with a real mobile robot and pedestrians. Within this application, the mobile robot is subjected to two kinds of interactions: firstly with the environment, through contact with the soil, and then with the proximity to surrounding obstacles; secondly the robot is subjected to Human-Robot interaction, which in this case must avoid at all costs.

Finally, in Chapter 4 are provided the conclusion remarks relative to the present dissertation.

Summarizing, in Table 1.1 are reported in a tabular form the main investigations and applications that will be covered in this dissertation and presented in the next chapters. Moreover, the table identifies the type of analysis that has been conducted (theoretical, numerical, or experimental) together with the type of interactions that occur, and the robot is subjected to in the specific considered applications.

	Activities				
Type of Interactions	A	B	C	D	E
Robot/Robot				✗	
Environment/Robot	✗	✗	✗	✗	✗
Human/Robot					✗
Type of Analysis					
Theoretical	✗	✗	✗	✗	
Numerical	✗	✗	✗	✗	✗
Experimental	✗	✗	✗		✗

Table 1.1: Summary of the main topics discussed in this dissertation. For each topic is marked the type of interaction encountered and the type of analysis that has been done.

Finally, the novelties and the contributions of this dissertation, by grouping them by the investigated application, can be listed as follows,

A. The contributions of the VRD-based lander investigation can be summarized in: the design and the numerical validation of a three-legged robotic lander designed for space applications; the leveraging of VRDs and cables for the synthesis of a mechanism of soft-landing which is designed to ensure a constant force acting on the pads; the experimental determination of the characteristics of the soil, and further implementations of a non-linear parallel spring damper contact model in which these parameters are used; and the introduction of three ratchets acting as a retention mechanism.

B. The contributions for the Archimede planetary rover investigation can be summarized as the development of a novel four wheels steering rover prototype, with articulated legs; the development of the firmware and software of the rover all based on ROS; the development of the dynamics model of the rover and numerical validation; the experimental validation of the numerical models; the development of an odometry sub-system for the rover and its numerical and experimental demonstration and accuracy evaluation, as well as an approach for joint limits avoidance based on the projection of the ICR and Ackermann steering; the development of ROS packages for high-level control as well as the modeling of the Gazebo simulation environment.

C. The contributions of the Towing Manipulation strategy investigation can be summarized in: the formulation and implementation of a quasi-static approach for the path planning of a towed object; the implementation of a continuous contact model; the development of dynamics models for this problem; the application of this approach to a four wheels steering planetary rover subjected to joint limits constraints and successive determination of the influence of its kinematics; and finally the experimental validation of the approach.

D. The contributions of the Tethered Mobile Robots investigation can be summarized in: the formal formulation of a kinematic analogy between a chain of tethered mobile robots and a redundant manipulator; the application of this framework for addressing a specific industrial problem, thus evaluating the feasibility of a chain of mobile robots and validating the developed framework; and finally the development of ROS packages for dynamic simulations in ROS-Gazebo and validation of the approach.

E. The contribution for the Crowd Navigation investigation can be summarized as the development of an extension of the model of moving crowds; the development of a functional Convolutional Neural Network based architecture to tackle this problem; the validation and comparison between the two chosen algorithms; the presentation of the developed parallel and asynchronous strategies to speed up neural network training speed; and finally the development of a ROS package for robot control and numerical and experimental validation of the algorithms and approach.

It must be pointed out that parts of this dissertation, are currently in consideration for publishing, or have been already published in scientific journals. These are:

M. Caruso, L. Scalera, P. Gallina, and S. Seriani, "Dynamic modeling and simulation of a robotic lander based on variable radius drums", *Applied Sciences*, vol. 10, no. 24, pp. 1–21, 2020.

S. Seriani, L. Marcini, M. Caruso, P. Gallina, and E. Medvet, "Crowded environment navigation with neat: Impact of perception resolution on controller optimization", *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 101, no. 2, 2021.

M. Caruso, P. Gallina, and S. Seriani, "On the modelling of tethered mobile robots as redundant manipulators", *Robotics*, vol. 10, no. 2, 2021.

N. Sesto Gorella, M. Caruso, P. Gallina, and S. Seriani, "Dynamically balanced pointing system for cubesats: Study and 3d printing manufacturing", *Robotics*, vol. 10, no. 4, 2021.

M. Caruso, L. Bregant, P. Gallina, and S. Seriani, "Design and multi-body dynamic analysis of the Archimede space exploration rover", *Acta Astronautica*, vol. 194, pp. 229–241, 2022.

M. Caruso, E. Regolin, F. J. C. Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, "Robot navigation in crowded environments: a reinforcement learning approach", *Preprints*, Dec. 2022.

M. Caruso, E. Regolin, F. J. C. Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, "Robot navigation in crowded environments: a reinforcement learning approach", Machines, vol. 11, no. 2, 2023.

M. Caruso, N. S. Gorella, P. Gallina and S. Seriani, "Towing an Object With a Rover", *IEEE Transactions on Robotics*, (in review).

Part I

Mobile Robotics for Space Applications

Chapter 2

Mobile Robotics for Space Applications

2.1 Introduction

Mobile robotics have found large use also within the space exploration applications field. This is a consequence of the fact that these environments are extremely hostile and inhospitable contexts for human beings. It follows the need of deploying robotic and autonomous systems in this field, which are able to operate and perform tasks that a human either it's impossible to do or that are too dangerous. After these considerations, many attempts and many pieces of research have been conducted in the past years on the development of these kinds of systems. These efforts have been focusing on the reliability, durability, and autonomy of these robotic systems.

As a consequence of all these research efforts, numerous space missions have been seen successfully launched, and their results have had major impacts on many fields of science, and some of these missions have been briefly described in Sec. 1.3.1. Within these missions, many different systems have been used and studied. Moreover, it could be seen the evolution of these systems and the technologies used, which are increasingly complex, within these space exploration missions.

Within these space exploration missions, it is possible to classify the robotic system used into two large main groups: the landers, and the rovers. The first one contains all those mobile robotic systems whose main purposes are the safe and soft landing and deployment of the payload and the scientific equipment on the surface of a celestial body. These can be fixed, and movable, thus having locomotion, disposable or reusable systems. Moreover, another important purpose is that they also serve as the basis for the other systems that have been deployed. The second group contains all those mobile robotics that possess locomotion and mobility, and are designed to cover large distances, even far away from the landing site, and therefore perform the various requires tasks, e.g. soil and rock sampling or taking images of the surroundings. All the robotic systems falling in both groups are very complex systems to study and develop because they need to be carefully designed to operate in unknown or partially known environments, which

are also unfriendly. Apart from the various environmental constraints and mission constraints, which have been described in section Sec. 1.3.1 and which these systems are subject to throughout their operational life, several interactions take place on these systems during their operations. Mostly these ones fall into the Environment-Robot classification described in Sec. 1.2. The most critical one is the interaction with the soil of the celestial body these systems need to operate, which in general is not ideal, it can be soft, and its properties are in general unknown and may vary along the celestial body surface.

For robotic landers, Environment-Robot interactions with the soil play an important role in the design process of the landing system. In fact, the characteristics of the soil and the contact models must be fully understood, and adequately described, in order to design a landing system for the robotic lander which grants: a safe and soft landing on the celestial body, controlled deceleration, and impulsive forces in order to prevent damages to the carried payload and instrumentation, stability to landing conditions perturbations. For the example of the walking landers, this interaction is not limited to ground impact only but occurs during the whole locomotion of the mobile robot. It follows, that for landers contact with the soil is not a drawback and unwanted phenomenon, but on the other hand, it is functional to the successful landing operations and deployment of the systems.

Also for the robotic rovers the soil interaction plays a fundamental role in the design of these systems, in particular in their locomotion systems. In fact, also for this case the characteristics of the soil and the contact models must be fully understood and described for the design process of the locomotion system, which must grant: adequate traction of the rover, high mobility on loose soils, withstands impacts occurring during traversing, and obstacle negotiation capabilities. It follows that, for robotic rovers, the soil interaction is again a functional phenomenon, which is needed for the locomotion of the mobile robot, in these environments. Interactions with the environment inevitably occur when the robot performs sampling operations.

Current iterations in mobile robots for space applications are dictated by the demand for the development of systems that are capable of covering greater distances in less time, in order to increase the investigative capabilities of the rover and the number of experiments that can be done during a single mission. One of the many goals is to transfer a well-known approach in the industry, which is that of collaborative multi-robot systems. Another goal toward which much attention and interest is being looked at, and this concept may be applied, is the creation of precursor and robotic space bases. In this context, multiple robotic systems are deployed on the surface of the celestial body and are intended to work collaboratively in order to build a precursor base before the arrival of human beings. This multi-robot system can be composed of several mobile robots, manipulators, and landers that are all connected to each other and work and perform operations together. A step in this direction was taken during the Mars 2020 mission, in which a rover, called Perseverance, and a drone called Ingenuity were deployed. It follows that in this context other interactions with the single robot will arise. These are the Robot-Robot interactions, between the robots

composing the multi-robot system and other Environment-Robot interactions. For the latter, it is apparent that, for the example of building a precursor base, the mobile robots must be capable of manipulating loads on the ground, thus another interaction arises.

Considering what has been described and mentioned above, this chapter aims to address some of the described issues and challenges that characterize the introduction of mobile robots in space applications. Specifically, in Sec. 2.2 will be presented a novel three-legged robotic lander prototype, whose soft-landing mechanism is based on the use of Variable Radius Drums (VRDs). In this section will be first provided an introduction and an in-depth state-of-the-art analysis specific to this problem. Subsequently, it will be described its design and modeling and the contact interaction with the soil will be addressed. The section will also describe the modeling of the lander in the ADAMS software package, and dynamic simulations will be conducted on two different landing conditions and finally validated the proposed robotic system.

In Sec. 2.3 will be presented a novel four wheels steering planetary rover prototype, called Archimede, which is completely based on ROS. The rover possesses a suspension system based on the use of articulated legs, which leverage the concept of the S-Structures. In this section will be first provided an introduction and an in-depth state-of-the-art analysis specific to this problem. Subsequently, it will be described its design, its software architecture which is based on ROS, and its dynamic modeling. Subsequently a planar kinematic and odometric subsystem, build on the former, will be presented. Moreover, a kinematics model, satisfying the Ackermann steering condition and an approach based on the ICR projection, used to fulfill kinematics constraints, will be presented. Subsequently, the modeling of the rover in ADAMS and Gazebo will be described. Tests both in simulations and in experimentation will be conducted in order to demonstrate and evaluate the developed odometry subsystem accuracy. Subsequently, dynamics simulations will be conducted, and then compared the results with the same experimental setting, thus validating the models, the rover, and finally the approach.

Finally, in 2.4 will be presented an in-depth investigation of the towing manipulation strategy of objects applied to mobile robots, and path planning of the towed object. Specifically, in this section, it will be first provided a description of the strategy, together with an introduction and an in-depth state-of-the-art analysis specific to this problem and to the general problem of the planar sliding problem. Subsequently, it will be described the modeling of the problem leveraging the quasi-static motion assumption. It will be therefore presented the approach for the path planning of an object being towed, together with the development of two dynamic models. Subsequently, the kinematics model of the Archimede rover and its control architecture will be introduced, in order to evaluate the effects of the kinematics constraints on the planned trajectories. Moreover, dynamics simulations are conducted in order to evaluate the deviation from the quasi-static assumptions. Finally, the experimental validation of the approach with the rover Archimede will be presented.

2.2 VRDs-based robotic lander

In this section, we aim to study and address an application belonging to the first macro group of robotic systems for space applications, which are the landers, and the related delicate soft-landing operation. Specifically, this section aims to present the design and study of an innovative three-legged robotic lander for space exploration. This robotic lander uses mechanisms based on VRDs that provide controlled deceleration during soft-landing operations. At the same time, this study aims to describe, model, and address the ground interactions that occur during the landing phase through the accurate modeling of ground contact patterns and their subsequent validation in simulated environments.

Indeed, in the case of robotic landers, the system is subjected to interactions through contact with the environment, through the lander's pads. It follows that this specific application falls into the Environment-Robot interactions group. It is well known that contacts are an issue when simulating MBS. However, even if they are an issue from a numerical perspective in this study contacts are functional phenomenons for the correct functioning of the robotic lander for two main reasons: the first is that contacts forces are needed in order to simulate landing operations; the second is that, as it will be seen, contacts are essential for the retention mechanism. Indeed, this mechanism uses a contact interference approach to lock the rotations of the leg.

Summarizing, the contribution of this sections can be listed in:

- The design and the numerical validation of a novel three-legged robotic lander intended for space applications;
- The use of VRDs and cables to create a soft-landing mechanism that ensures a constant force acting on the pads and the introduction of ratchets as retention mechanisms which allows greater stability;
- The experimental determination of the soil's properties, and successively used in the implementations of a non-linear parallel spring damper contact model for the numerical methods.

2.2.1 Introduction

In space exploration, the task of landing on a planet, an asteroid, or any other celestial body is crucial and can decide the fate of the entire operation. Many missions in the last decades have failed as a result of the lander crashing to the ground during this sensitive operation. Despite these unsuccessful missions, numerous others were successful, and numerous systems were tested and used throughout the years to ensure the soft landing of a planetary lander.

The goal of the most notable mission, Apollo 11, from the NASA Agency was to send a probe carrying humans to the Moon's surface for the first time. A powered descent maneuver was first used to achieve a successful landing. Then, a suspension mechanism with a honeycomb shock absorber on each lander-leg strut enabled a successful touchdown [144]. Two twin probes, whose name are Viking 1 and Viking 2, were dispatched to Mars as part of the NASA's Viking Project,

which was launched in 1975. The two landers used legs and adopted a combination of rocket propulsion and a parachute to achieve a smooth touchdown on the planet's surface [145]. The same system has been used, in 1997 in the lander of the Cassini-Huygens NASA-ESA-ASI mission. The lander successfully achieved soft-landing on Titan, one of Saturn's moons. Following the Viking mission, NASA launched further probes to explore the red planet. To begin with, in 1996, the NASA Pathfinder mission began operations [146]. Subsequently, still from NASA, the Mars Exploration Rovers (in short MERs), also commonly known as Spirit and Opportunity, have been sent to the planet in 2004. Soft-landing, for all these three missions, has been achieved through an approach that is based on the usage of gas-filled airbags [147]. Later on, in 2011, another NASA's noteworthy mission was the Mars Science Laboratory (in short MSL). In this mission, the planetary rover named Curiosity has been deployed on the surface of Mars. Using a combination of a parachute, rockets, and the Sky Crane technology, a successful landing was accomplished. In more recent times, the ESA's Rosetta mission employed a powered descent to achieve soft-landing on the comet and an anchoring mechanism in order to stay attached to it [148]. The Soviet Space Program deployed various advancements concurrently with NASA during the Space Race competition, beginning with impactors [149] and leading to the lander Luna 9 (February 1966), which has gone down in history as the first probe achieving successful soft-landing on another celestial body [150]. On the other hand, still from the Soviet Space Program, in 1970, the planetary rover Lunokhod 1 was the first rover to successfully land on another celestial body [151]. More recently, the China National Space Administration (in short CNSA) began a space program (Chinese Lunar Exploration Program) in the 1990s that allowed the Chang'e 3 lunar lander and the rover Yutu, thus becoming the first Chinese lunar planetary rover, to soft-land in December 2013 on the Moon [152]. Subsequently, belonging to the same program, came the Chang'e 4 lander in January 2019 [153]. Numerous researchers have been posing much effort into studying, in various study domains, the issue of lander's soft-landing operations in the last decades, also to improve the efficiency of the landing systems. For instance, Wang et al. in 2019 investigated an approach that relies on semi-actively controlling the damping force of the magnetorheological fluid dampers in order to adapt to the landing scenario [154]. Additionally, through the utilization of numerical optimization methods, it is possible to conduct the design process of the shock absorbers [155] as well as optimize the stability of the lander [156]. Additionally, more versatile spacecraft, such as walking landers have been investigated in [66, 67].

When it comes to actual and concrete planetary missions which involve the use of landers, hence integrating a soft-landing system, the issue of soft-landing operations must be investigated very carefully. Moreover, all possible operating scenarios must be taken into account and the device must be as reliable as possible since even the smallest setback can turn into a disaster. When dealing with the issue of soft-landing many problems can be addressed. However, the major ones are the ones related to the stresses the lander is subjected to. Indeed, a robotic lander during his operational life is subjected to all kinds of interactions, and the device must be resistant to them. One of these interactions, which is

extremely critical, is the interaction that occurs when the lander contacts the underlying soil in the phase of touchdown. Indeed, landing is a process that may be very fast, and during this process, excessive stresses and accelerations can be exchanged. Having said that, therefore the modeling of the dynamics of the lander as well as the modeling of the interactions which occur between the beneath soil and the lander are the two key issues that may be tackled in the investigation of soft landing operations. The ground properties of celestial bodies have a significant impact on the mission [157], and as a result, several attempts are undertaken to predict them in beforehand [158, 159]. Numerous publications describe the characterization of the soil, especially with regard to its granulometry [160]. Summarizing, the interactions to which these robotic systems are subjected are mostly with the surrounding environment. However, given the big push concerning the development of autonomous systems, and systems composed of multiple robots, in the context of a robot precursor base; in principle, there can be multiple systems cooperating. Thus, Robot-Robot interaction is also possible.

With the advent of commercial simulation software, several researchers attempted to investigate the soft-landing issue by using them. Most of the studies found in the literature see authors selecting a lander having four legs as a case study. For instance, Xu et al. (2011) showed that landing simulations, for a lunar lander, can be done utilizing MSC-ADAMS modeling as a rigid body the beneath soil; and MSC-DYTRAN can be utilized when considering flexible the soil body [161]. Using the finite elements technique together with a non-linear transient dynamics methodology, Wan et al. (2010) selected MSC-DYTRAN for the soft-landing simulations of a lunar lander [162]. Furthermore, utilizing LS-DYNA and ABAQUS, other commercially available software packages, Zheng et al. (2018) used explicit finite element methods to simulate the impact of a lander [163]. Another example of a study conducted utilizing ABAQUS, implementing an explicit non-linear finite element method, for soft-landing purposes, is provided by the work of Liang et al. (2011). In their research, the interaction between the lunar surface and a lander utilizing aluminum honeycomb shock absorbers [164]. Moreover, other researchers investigated the lander's impact using MSC-ADAMS, by modeling the soil either as a rigid or an elastic body [98–102].

The notion introduced by Seriani in [165] is carried over into modeling of the dynamics of a three-legged robotic lander and relative simulations in this investigation. The lander is based on the utilizing of Variable Radius Drums (VRDs). The suggested mechanism is a non-reversible device that permits a landing body to slow down during the period of impact with the soil. As stated by Seriani and Gallina in [166], a VRD is a device distinguished by the fluctuation of the drum radius along its profile. It is utilized in this problem to guarantee a constant force, therefore a structure's controlled deceleration while landing. VRDs were investigated for a variety of uses, such as to steer loads down horizontal paths, improving a legged robot's locomotion [166] or using them in gravity-compensation systems [167].

In 2018, Scalera et al. investigated a cable-based robotic crane powered by VRDs that could transport a weight over a flat working surface [168]. Addition-

ally, in a cable-suspended robot, Fedorov and Birglen (2018) used two antagonistic VRDs to direct the end-effector along a predetermined pick-and-place trajectory. The static balance of a pendulum was also done using the same idea [169]. The purpose of using the VRD in these situations is to produce geometric trajectories in a mechanism. In contrast, within this study, a synthesized VRD is utilized to produce a particular force when coupled to a linear preloaded spring device. Other uses of VRDs are cable drives in soft robot joints[170], upper limb orthoses [171], energy-saving devices [172] and force control [173].

Within this investigation, an experimental method has been leveraged to determine the soil's properties, which is a basaltic sand having volcanic origin with an average particle size range of 3–5 mm, and that is classified into the coarse sands as stated by NASA [174, 175]. This has been done in order to build a solid baseline for the numerical simulations and obtain meaningful results.

This work presents several novelties in comparison to relevant literature, particularly the work of Seriani [165], which implements an ideal linear spring-damper contact and an ideal Coulomb friction model: it provides an experimental determination of the properties of soft basaltic sand soil for implementation in a spring-damper ground contact and implements a non-linear stiffness model; additionally, the used contact model implements experimentally defined static and dynamic friction coefficients. Additionally, the theoretical model's numerical implementation is more stable, preventing the need for numerical damping of the legs' rotation. Additionally, three ratchets are used in this work's 3D lander as retention mechanisms to make the system irreversible. Their primary function is to prevent the lander from rebounding after landing by restricting the legs' capacity to rotate backward once they have fully rotated and the payload has entirely decelerated. It also prevents rotation altogether, keeping the lander's center of mass nearer the ground and enhancing the lander's stability. Additionally, the ratchets devices prevent the preloaded springs' energy from discharging, preventing the lander from bouncing excessively.

This section is structured as follows: in Sec. 2.2.2 is reported the design and the modeling of the VRD mechanism, the derivation of the bi-dimensional dynamics model of the lander, together with a detailed description of the soil contact models elected; in Sec. 2.2.3 is reported and detailed described the modeling process of the lander in the ADAMS simulation software package; in Sec. 2.2.4 are presented the simulation test cases and the corresponding obtained results are obtained; finally, in Sec. 2.2.5 the results are discussed and compared to the theoretically based model.

2.2.2 Dynamic modeling of the VRDs-based lander

The concept presented in [165] provides the foundation for the lander construction that has been investigated in this paper. Additionally, as the lander's structure begins to ascend at the conclusion of the deceleration phase, a ratchet mechanism is added to prevent the counter-rotation of the legs. According to Figure 2.2, the lander is made up of a body, the main structure, having an hexagonal shape on which the payload is placed, and three legs that are oriented at an angle of $2/3\pi$

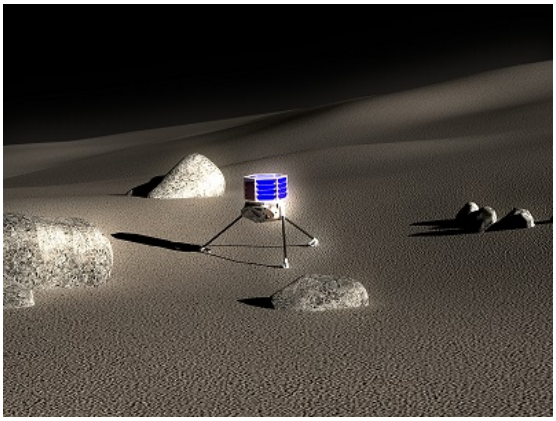


Figure 2.1: 3D Graphical Rendering of the three legged Variable Radius Drum robotic lander

from one another and have a fixed offset from the center of mass of the lander. In this paragraph is first presented the modeling of the VRD mechanism, and subsequently the dynamic modeling of the lander.

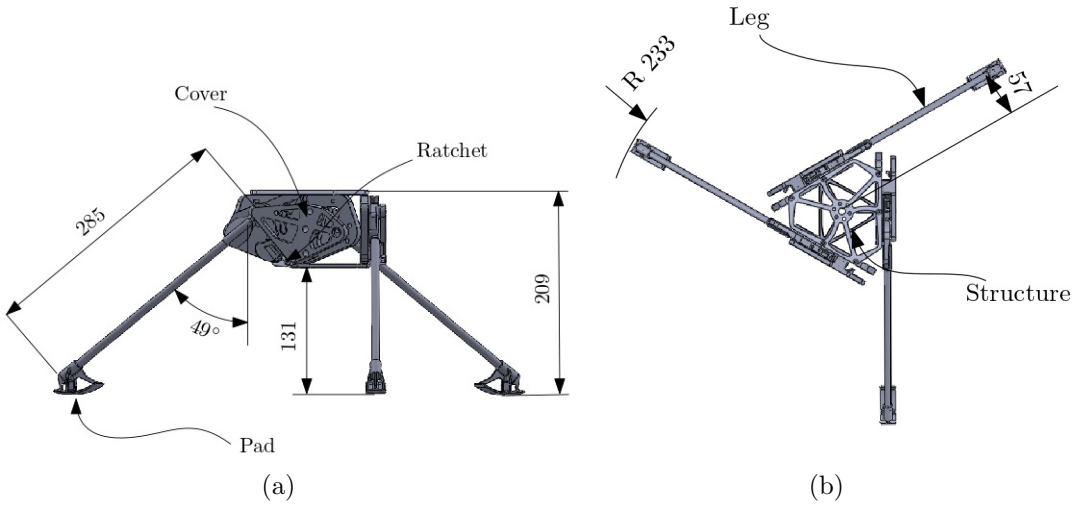


Figure 2.2: The CAD model of the proposed three-legged lander, along with labels for its main parts and main dimensions: (a) lateral view; (b) top view [176]

In Figure 2.1, on the other hand, is reported a 3D graphical rendering of the three-legged robotic lander in a lunar-like environment, based on the use of VRD, that is studied and presented in this section. In Figure 2.3 is reported a schematic representation of the single leg system model of the lander just when impact occurs.

Modeling of the Variable Radius Drum mechanism

Consider the lander leg mechanism shown in Figure 2.3. A fixed radius pulley, (FRP for short) with radius r_1 attached to the leg, a VRD, a FRP with radius r_2

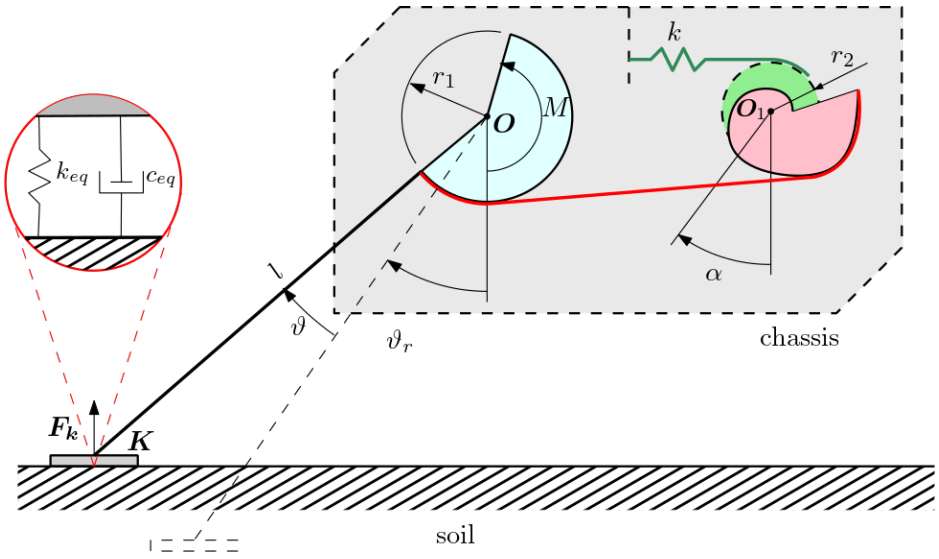


Figure 2.3: Schematic view of the lander-leg mechanism at the beginning of the impact with soil with a detailed view of the model of the normal contact force as a spring-damper system [176]

.

linked to the VRD, and a preloaded spring make up each lander-leg mechanism's major components. Additionally, the system makes use of two cable elements. The first cable's extremities are fastened to the FRP and the VRD, respectively (red line). The extremities of the second cable (green line) are fastened to the FRP (rigidly linked to the VRD) and the preloaded spring, which is in turn linked to the lander chassis. The VRD can mold the preloaded spring's elastic response thanks to its particular configuration. More specifically, this translates into the result that the FRP attached to the leg experience a reactive torque M produced by the preloaded spring and shaped by the VRD. It is therefore possible to model and approximate the induced torque M by a continuous and differentiable function as follows [165, 177, 178],

$$M = \frac{2}{\pi} M_0 \arctan\left(f\vartheta\right) + k\vartheta \tag{2.1}$$

where k represents the spring stiffness, f an approximation factor which models the "hardness" of the step function, M_0 represents the preloaded torque, and finally ϑ represents the relative rotation of the lander's leg from its rest position ϑ_r.

Consider now the design requirements, which have been already mentioned in the introductory paragraph of this section. Specifically, in this study it is required that the lander impacts the underlying soil with a controlled force, thus a controlled deceleration of the structure and the payload. Therefore, by imposing the contact force $\boldsymbol{F_k}(\vartheta) = c$, i.e. a constant force the lander's pad is subjected, then it is possible to synthesize the VRD profile as explained in [166, 168]. Moreover, a generic point of the VRD's profile, we will denote with $\boldsymbol{P_m}$, can be expressed as follows,

$$\boldsymbol{P_m} = \boldsymbol{T}(\alpha) \begin{Bmatrix} c_d \\ 0 \end{Bmatrix} + \boldsymbol{T}(\alpha)\boldsymbol{T}(-\gamma)\boldsymbol{T}\left(-\frac{\pi}{2}\right) \begin{Bmatrix} l_t \\ 0 \end{Bmatrix} \tag{2.2}$$

It is important to point out that the point $\boldsymbol{P_m}$ is expressed in the local frame of the VRD. Specifically, the terms in Eq. (2.2) are the relative rotation of the VRD α; l_t which represent the distance between the point $\boldsymbol{P_m}$ and the idle pulley center $\boldsymbol{O}$; and $\boldsymbol{T} \in \mathbb{R}^{2\times 2}$ which in this context indicates a generic rotation matrix.

The distance l_t is defined as follows,

$$l_t = \frac{c_d \sin\gamma}{1 + \frac{\frac{d^2 g}{d\alpha^2}}{\sqrt{c_d^2 - \left(\frac{dg}{d\alpha}\right)^2}}} \tag{2.3}$$

where $g(\alpha)$ is referred to as the cable wound function; c_d instead indicates the distance between the points $\boldsymbol{O}$ and $\boldsymbol{O_1}$; finally γ represent a parameter which is defined as follows,

$$\gamma = \arccos\left(\frac{1}{c_d}\frac{dg}{d\alpha}\right) \tag{2.4}$$

It follows that with reference to Figure 2.3 is it possible to establish the expression of the cable wound function that is,

$$g(\alpha) = r_1 \vartheta \tag{2.5}$$

The first and second derivatives of Eq. (2.5) with respect to α are immediate. On the other hand, defining $\boldsymbol{F_s}$ as the elastic force generated by the preloaded spring, and by applying the principle of virtual works it follows that,

$$\frac{d\vartheta}{d\alpha} = \frac{||\boldsymbol{F_s}||r_2}{||(\boldsymbol{K} - \boldsymbol{O}) \times \boldsymbol{F_k}||} \tag{2.6}$$

These relations are fundamental because they allow the synthesizing of the VRD profile for the case of constant pad-soil force during the lander's impact with the soil.

Lander's dynamics model

Due to the symmetry of the robotic lander, it is possible to derive the dynamics model of the lander by considering only one-third of the lander. Consider the model to be bi-dimensional and the chassis of the lander constrained to move only along the vertical axis. Moreover, the model is considered in free falling, thus the constraint given by the soil is not included. It follows, that the system is composed of two bodies and is a 2 DOFs model. It is possible therefore to define the Lagrangian function as $\mathcal{L} = \mathcal{K} - \mathcal{U}$, where $\mathcal{K}$ and $\mathcal{U}$ are the kinetic and potential energy respectively. The EOMs of the system can be finally derived by using the Lagrangian approach,

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \sum_{i=1}^{n} Q_{c,i} + \sum_{j=1}^{l} Q_{n.c,j} \tag{2.7}$$

where $Q_{c,i}$ and $Q_{n.c.,j}$ are the i-th conservative and j-th non-conservative generalized forces, respectively, which has not included in the Lagrangian function. It is recalled that, in general, for a generalized coordinate q, an external force $\boldsymbol{F_c}$ with application point $\boldsymbol{x_c}$, the resulting generalized force is expressed as follows,

$$Q_{c,q} = \boldsymbol{F_c} \cdot \frac{\partial \boldsymbol{x_c}}{\partial q} \tag{2.8}$$

The contact force exchanged between the soil and the lander's pad when contact occurs is introduced in the system as an external force applied to the system.

Contact models and soil carachterization

In this investigation, the robotic lander is a MBS, which is subjected to multiple contacts exchanged with the underlying soil. It follows that effects coming from contact dynamics are introduced in the system. It is, therefore, necessary to introduce into the system, models which manage contacts in order to account for their effect. Therefore, it has been decided to utilize one of the penalty methods that has been introduced in Sec. 1.4.2. More precisely it has been chosen to use a model which describes the vertical contact force as a non-linear spring-damper system, as explained in [156], and that is expressed as follows,

$$\begin{cases} F_k = -k_{eq}x^{\delta} - c_{eq}\dot{x}, & x \leq 0 \\ F_k = 0, & x > 0 \end{cases} \tag{2.9}$$

The terms in Eq. (2.9) are δ which represent exponent factor accounting for the non-linearity of the elastic component, k_{eq} which indicates the equivalent spring stiffness, c_{eq} defined as the equivalent damping factor, while x, in this case, represent the penetration of the lander's pad into the soil and $\dot{x}$ its time derivative.

The tangential component of the contact force, i.e., the frictional component is modeled by utilizing a simple Coulomb friction model, and that is expressed as follows,

$$\begin{cases} F_f = -\mu_s N, & \dot{y} = 0 \\ F_f = -\mu_k N, & \dot{y} \neq 0 \end{cases} \tag{2.10}$$

In Eq. (2.10) the variable $\dot{y}$ represents the relative speed of the contacting surfaces, N represents the normal force between the surfaces, while μ_s and μ_k represent the static and dynamic friction coefficients respectively.

It is well known that one of the drawbacks of the penalty models for contact dynamics is the correct choice and estimation of the contact parameters. Therefore, it is required to estimate the parameters of Eq. (2.9) and Eq. (2.10), in order

to perform accurate and meaningful comparable simulations. For this purpose, it has been chosen a sample of basaltic sand of volcanic origin with an average grain size ranging from 3 to 5 mm, for the determination of the soil characteristics. The first step in this direction consisted of performing some experimental tests in order to estimate the module of subgrade reaction k, which is defined as follows,

$$k = \frac{q}{\Delta} \tag{2.11}$$

This modulus and the soil equivalent stiffness k_{eq} are directly related. In Eq. (2.11), q represents the pressure acting in the contact patch, while Δ indicates the penetration depth.

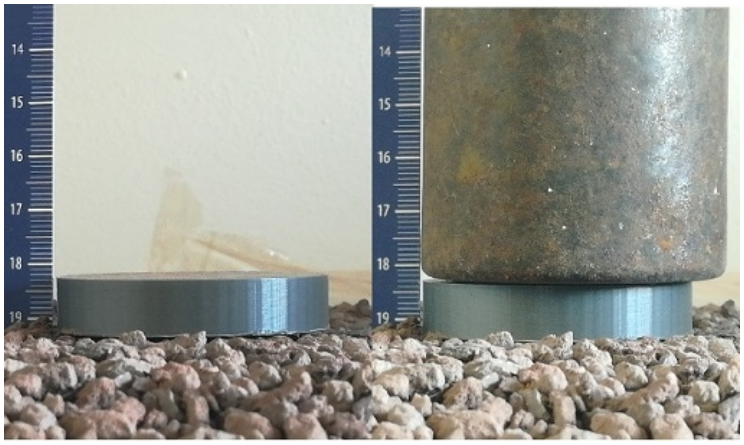
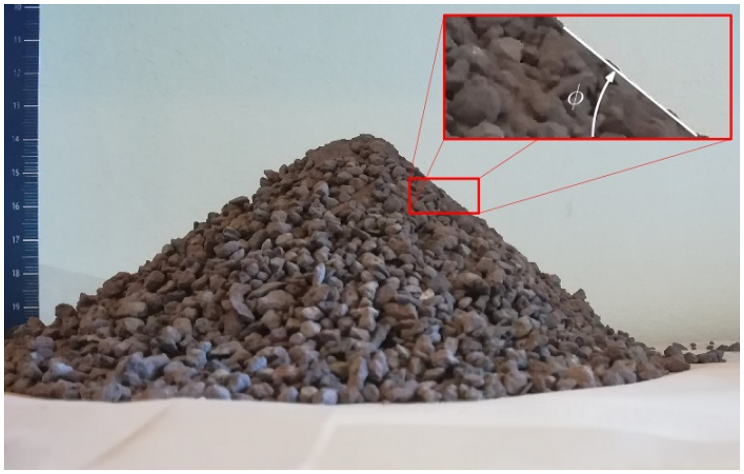


Figure 2.4: Experimental measure of the penetration depth [176]

In order to execute the estimation tests more accurately, a circular plate with dimensions similar to the lander pad is placed on the surface of the soil sample. Therefore, the deflection under static conditions is measured after a 1 kg weight is placed above the circular plate. As the deflections are extremely small and sub-millimeter, the measurements have been conducted via image analysis. Specifically, the deflection is estimated through a comparison of the images taken before and after applying the load on the plate. This procedure can be better seen in Figure 2.4, where the scale unit is centimeters. These tests have been executed for a total of 13 times in order to have better consistent measurements. It has been estimated the penetration $\Delta = (0.141 \pm 0.049)$ mm, therefore by leveraging Eq. (2.11) the equivalent stiffness of the soil can be computed with suitable geometric considerations [100]. Specifically, within this study, it has been estimated $k_{eq} \approx 70\,\mathrm{N\,mm^{-1}}$. For what concerns the equivalent damping factor, reasonable values usually are $0.1 \div 1\%$ of the k_{eq} parameter [100]. Specifically, within this study, it has been elected to use $c_{eq} = 0.7\,\mathrm{N\,s\,mm^{-1}}$.

For what concerns the estimation of the friction properties, in a preliminary approach, the static friction coefficient may be calculated as $\mu_s \approx \tan(\phi)$, where ϕ is referred to as the internal friction angle of the soil. A number of 15 experiments have been done and the angle has been empirically measured. These tests are executed by spilling a certain amount of the sample soil on the surface, and



(a)

Figure 2.5: Experimental characterization of the soil: (a) measure of the internal angle ϕ [176]

subsequently, a photograph of the vertical profile of the heap is then taken. The internal angle of the soil ϕ is therefore calculated, as indicated in Figure 2.5 and estimated as $\phi = (38.82 \pm 2.02)^\circ$. It follows that $\mu_s \approx 0.804$, while μ_k is assumed as $\mu_k = 0.7\mu_s$.

Table 2.1 provides and summarizes the lander design parameters considered in this study, where m is the mass of the chassis, m_a is the mass of the leg, g is the gravitational acceleration, H is the drop height, k_s the spring stiffness, l_c is the length of the lander's leg up to the pad's center, while l_m is the maximum length of the lander's leg, i.e. up to the pad's tip.

Table 2.1: Parameters used in the design of the lander.

Parameter	**Value**	**Parameter**	**Value**
r	5.00×10^{-2}m	g	$9.81\,\mathrm{m\,s^{-2}}$
r_1	3.20×10^{-3}m	H	0.450 m
m	0.500 kg	f	1.00×10^4
m_a	3.70×10^{-2}kg	M_0	1.54 N m
l_c	25.3 cm	l_m	27.1 cm
k_s	$50.0 \times 10^3\mathrm{N\,m^{-1}}$	Material	PLA

2.2.3 ADAMS modeling

This section describes the setup for the numerical simulations, in addition to the modeling of the MBS that represents the robotic lander, that is conducted in ADAMS. Figure 2.7 shows the outcome of this modeling process, in which the soil is modeled as a rigid body having a box shape and the whole robotic lander as an MBS. Moreover, each rigid body that is necessary for the proper functioning of the whole system is connected to the lander MBS model by using the proper

joints. These components are the lander's hexagonal structure (chassis), the three legs, the three ratchet mechanisms, and the three covers. On the other hand, the parts of the lander which do not contribute to the functioning of the whole system, as well as the payload, are taken into account by including their mass and inertia attributes in the lander's chassis. It is useful to mention that, in the simulation environment, the joints are considered ideal ones, therefore no deformations or friction or damping, or end-stops are taken into account.

The components such as the lander chassis (yellow), the legs (gray), the ratchets (red), and the covers (green), can be better seen in a graphical form in Figure 2.6 and Figure 2.7.

In the ADAMS simulation environment created, it has been implemented a series of external forces. These include the gravitational force field, a torsional spring on the joints between the ratchet mechanism and the chassis, and an applied nonlinear torque between each leg and the chassis. The torsional spring between the ratchet and the chassis is introduced in order to imitate the behavior of the ratchet mechanism. More precisely the ratchet, for it to function properly, must be free to deform elastically under the action of the contact forces induced by the paw. In light of this, the ratchet is considered as a cantilever that deflects in response to the contact occurring between the leg and the ratchet in order to estimate values that characterize the approximating torsional spring introduced in the model. Therefore, according to the elastic theory, it is possible to derive the equivalent torsional stiffness, and defined as follows,

$$k_{r,eq} = \frac{E_r I_r}{l_r} \tag{2.12}$$

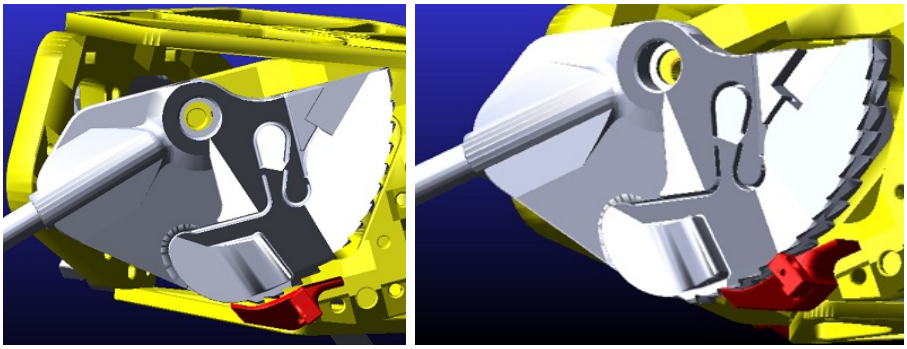
where I_r is the moment of inertia of the resistant cross-section of the ratchet, E_r is the Young elastic modulus of the PLA, and l_r is the length of the cantilever, therefore it has been possible to estimate this coefficient as $k = 288\,\mathrm{N\,mm\,rad^{-1}}$.

In order to replicate the resistance torque imparted to the leg joint caused by the motion of the VRD pulley, as described in Sec. 2.2.2, a non-linear torsional spring is placed among each leg and the chassis. Instead of utilizing Eq. (2.1), within the simulated environment in ADAMS, a cubic interpolating polynomial function, whose coefficients are reported in Table 2.2, has been used to simulate the reaction of the VRD pulley.

Table 2.2: Coefficients used for the cubic interpolating polynomial approximating the VRD pulley response

Coefficient	Value
a_0	1147
a_1	18.6761
a_2	-0.2175
a_3	-1.0823×10^{-4}

In addition to the definition of the external forces, for the proper functioning of all the parts of the robotic lander also a set of contact forces had to be defined



(a) (b)

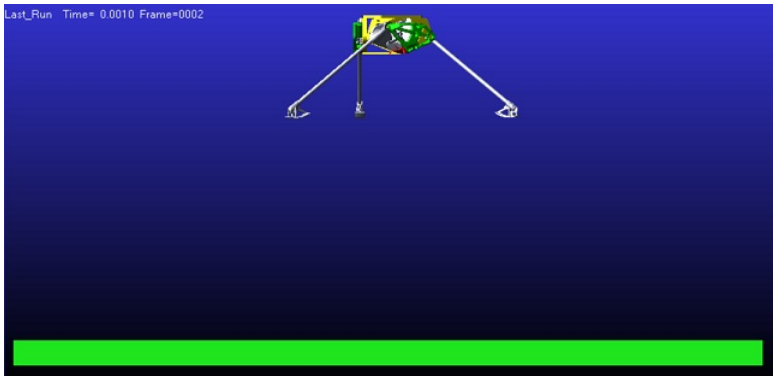
Figure 2.6: View of the leg joint and the ratchet mechanism used in ADAMS. (a) The ratchet and the leg joint at rest position; (b) the ratchet blocking the opposite rotations of the leg thanks to the contact force between the ratchet and the leg [176].

and introduced in the model. It is worth remarking that every contact occurring in this mechanical system is not unwanted but rather functional, and it will be seen how. These forces are:

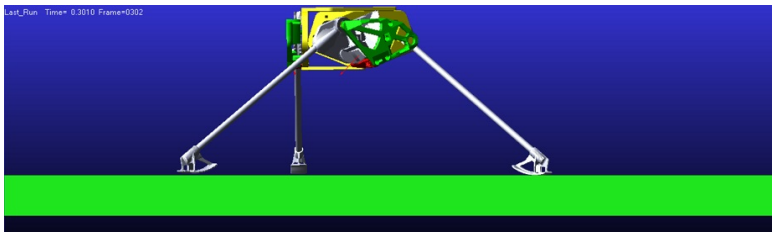
- **Contact between the leg and the soil**: This kind of force has been introduced for each of the three legs of the robotic lander and the soil. These are the most important ones since they are functional for the soft-landing simulations. The stiffness coefficient, nonlinear force exponent, damping coefficient, penetration depth, and static and dynamic friction coefficients are the variables that need to be specified for this contact force. The set of parameters used is then the ones that have been estimated in Sec. 2.2.2.

- **Contact between the leg and the cover**: This kind of force has been introduced between each leg and the contiguous cover's body. With reference to Figure 2.6, it can be seen that a small geometry is modeled on the side surface of the leg. This geometry, when the leg of the lander is closing, goes in contact with the cover. This has the advantage of restricting the joint from rotating freely; hence this element is functional to the simulation because it introduces a needed physical lower end-stop thanks to the contacts defined between the legs.

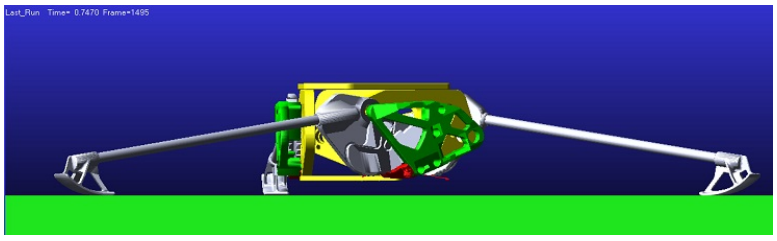
- **Contact between the ratchet and leg**: This kind of contact has been introduced between each leg and the adjacent ratchet body. For the retention mechanism to operate properly, the defining of contact forces between the leg and the adjacent ratchet is crucial. Particularly as the structure slows down during touchdown, the leg rotates in the permitted direction and the force of this motion causes the ratchet to bend, causing the leg to rotate. Instead, the tip of the ratchet mechanism engages with the tooth on the side of the leg joint when the structure stops decelerating and begins to move upward, enabling the ratchet to restrict the reverse rotation of the leg joint. This is evident right away from Figure 2.6a and Figure 2.6b.



(a)



(b)



(c)

Figure 2.7: View of the lander MBS and the soil modeled in ADAMS: (**a**) the initial condition of the simulation; (**b**,**c**) the lander configuration at the beginning and at the end of an impact simulation on a flat surface, respectively [176]

Every contact force introduced is described as a solid-to-solid contact force, consisting of a frictional force according to the Coulomb model Eq. (2.10) and a normal contact force according to Eq. (2.9).

Finally, Table 2.3 summarizes the geometrical and physical properties of the functional parts of lander MBS, while Table 2.4 summarize the kind of joints that have been used between the single parts.

2.2.4 Simulation results

The results of the numerical simulations on the three-legged robotic lander prototype model are reported within this section. The multi-body software package MSC ADAMS 2019 has been used in order to simulate the system's dynamic behavior. In this investigation, two simulated instances are taken into account:

(*Case* 1.) **Simulation of an impact on a flat surface:** in this simulation, the behavior of a lander free-falling from a set height and let colliding with

Table 2.3: Geometrical properties of the main functional rigid bodies composing the lander MBS model. All the inertia components are expressed in kg mm^2 [176].

Bodies	$\boldsymbol{m}$ [kg]	$\mathbf{I}_{\boldsymbol{xx}}$	$\mathbf{I}_{\boldsymbol{yy}}$	$\mathbf{I}_{\boldsymbol{zz}}$	#
Structure	0.5	430.14	684.78	430.14	1
Leg	3.7×10^{-2}	374.29	10.67	364.37	3
Ratchet	5×10^{-4}	9.40×10^{-2}	0.11	3.49×10^{-2}	3
Cover	1×10^{-3}	17.00	11.49	6.45	3

Table 2.4: Joint type used between each pair of rigid bodies.

First Body	**Second Body**	**Joint Type**	**Physics**	#
Structure	Leg	Revolute	Ideal	3
Structure	Ratchet	Revolute	Ideal	3
Structure	Cover	Fixed	Ideal	3
Soil	Ground	Fixed	Ideal	1

a horizontal flat surface is investigated. Within this test attention is posed on the distribution of contact forces, the lander's deceleration, the leg's angular displacement, and the effects the ratchets introduce.

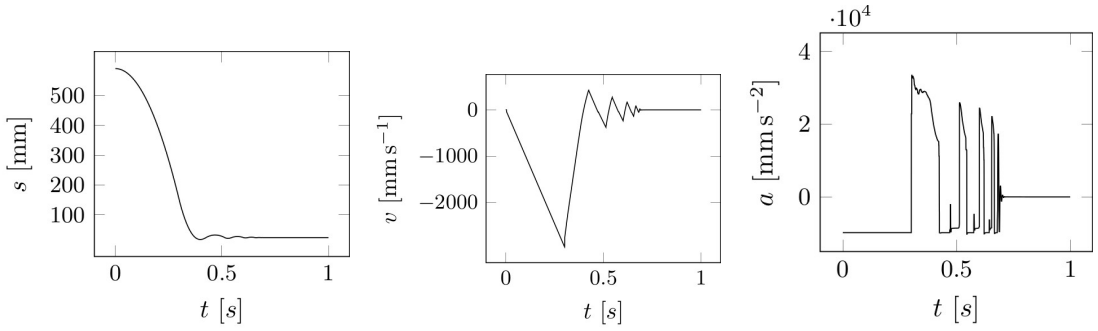
(*Case* 2.) **Topple simulation:** In this simulated scenario, the behavior of the lander colliding on an inclined terrain, following a fixed height free-fall, is explored. The critical angle that causes the lander to topple is then determined.

Case 1. Simulation of an impact on flat surface

In this simulated scenario, the robotic lander is allowed to fall freely from a predetermined height before colliding with the underlying box that serves as the soil model. The starting, impact moment, and ultimate configurations of the robotic lander MBS in this simulated scenario are shown in Figure 2.7, respectively. In this study, a trial-and-error approach is used to determine the highest allowable drop height while taking into account that the lander's payload is 0.5 kg. Additionally, the most representative findings for the maximum drop height, which is determined to be $H = 0.45$ m, are provided within this paragraph. One second has been chosen as the representative simulation time, since it is long enough to capture the transient phenomenons that occur between the simulation's start and the lander MBS stabilization.

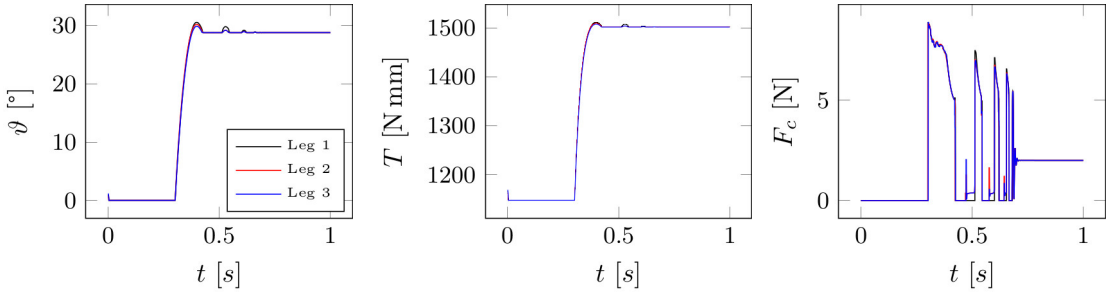
The lander's MBS center of mass displacement, velocity, and acceleration along the vertical axis—the axis of motion of the free-falling lander—are shown in Figure 2.8a–c), respectively. Figure 2.9a–c) shows instead the rotations of the joints in each leg structure, the torque produced by each of the three VRDs, together with the forces that each pad applied to the ground during the soft-landing simulation. Last but not least, Figure 2.10a,b) displays the rotations of

every ratchet-structure revolute joint as well as the evolution of the lander MBS's kinetic energy, respectively.



(a) (b) (c)

Figure 2.8: Information about the state of the lander's MBS along the vertical axis: (a) the displacement of the center of mass; (b) the velocity; (c) the acceleration [176].

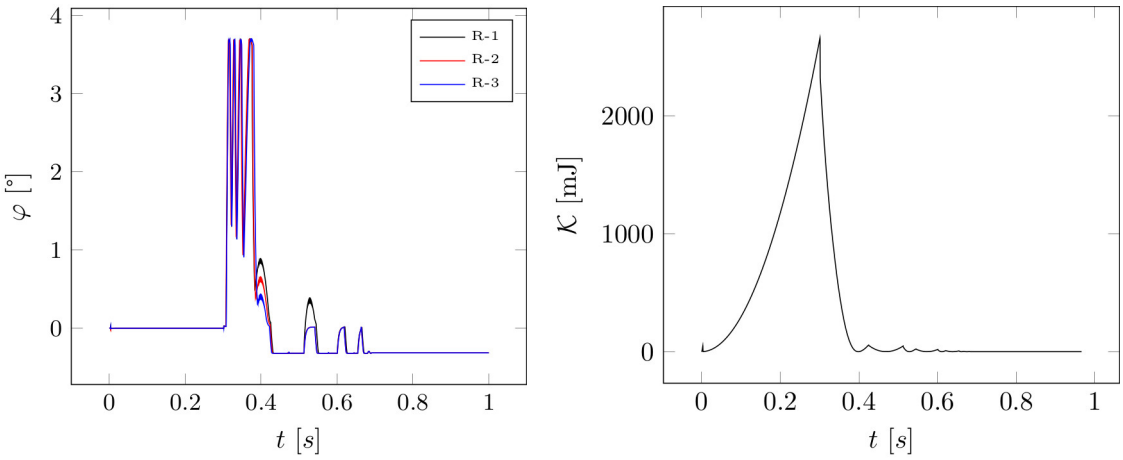


(a) (b) (c)

Figure 2.9: (a) The curves representing the angular rotation of each of the three leg structure revolute joint; (b) the curves representing the torque response of each of the three VRDs; (c) the curves representing the contact force between each of the legs with the soil [176].

Case 2. Topple simulation

Simulations were carried out to study the lander toppling, and the results are reported in this paragraph. The specific purpose of the numerical experiments is to determine the angle at which the lander will topple, which is then referred to as the landing critical angle β_{cr}. It is worth specifying that the same model from the previous study is taken into consideration in order to conduct the rollover simulations. Furthermore, in order to maintain a constant drop height between each simulation, as rotated by increasing angle around the z axis, the ground model is rigorously shifted along the y axis. To ensure a safe soft-landing without the lander tipping, a range of slope angles that are acceptable can be investigated using the ADAMS simulations for this case. It is crucial to note that the same

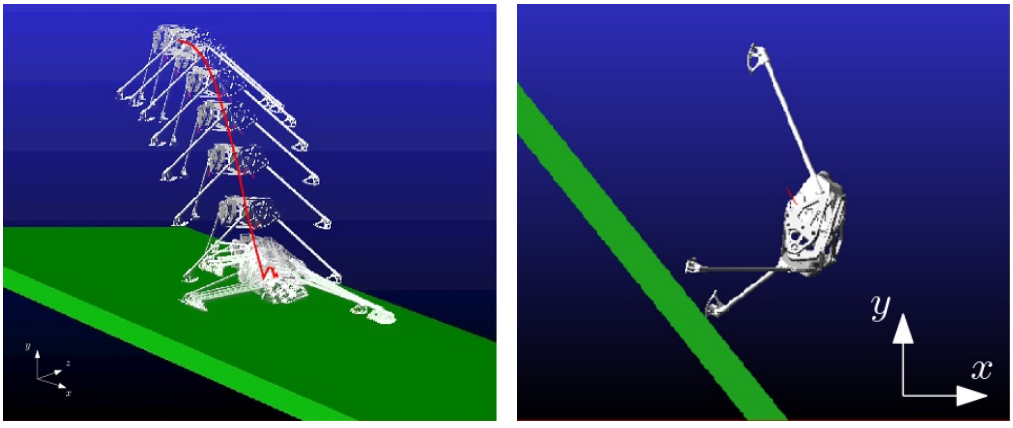


(a) (b)

Figure 2.10: (a) The rotation evolution of each of the ratchet-structure joint; (b) the kinetic energy evolution over time of the lander MBS [176].

maximum drop height, $H = 0.45\,\text{m}$, was used for all the simulations. Following this course of action, it is discovered that the lander topples at a key landing angle of 51.6°, as shown in Figure 2.11. On the other hand, in Figure 2.12 are shown some extracted frames, forming a storyline, of a sample topple simulation using a slope angle greater than the critical one.

Additionally, some significant findings for the test case of a simulation of the soft-landing on an inclined surface with a 1:5 inclination are provided.



(a) (b)

Figure 2.11: (a) A time-series representation of the simulation for the lander MBS soft-landing conducted on an inclined surface; (b) the lander MBS toppling at the landing critical angle β_{cr} [176].

Meaningful quantities, such as the lander's center of mass displacement, velocity, and acceleration are shown in Figure 2.13a-c, respectively. On the other hand, quantities such as the rotations of each joint in the leg structure, the torque produced by every one of the three VRDs, as well as the contact forces between

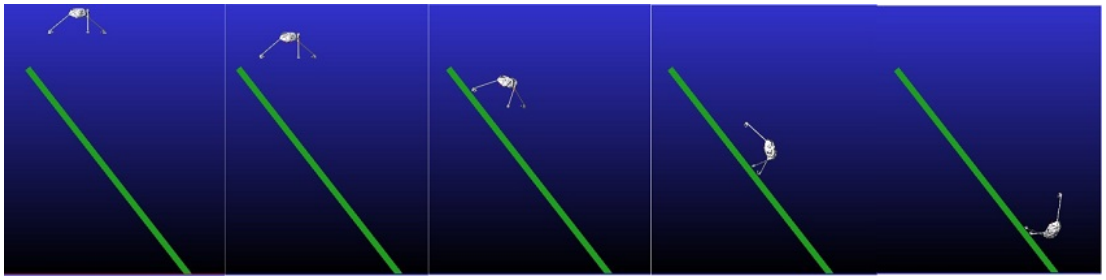
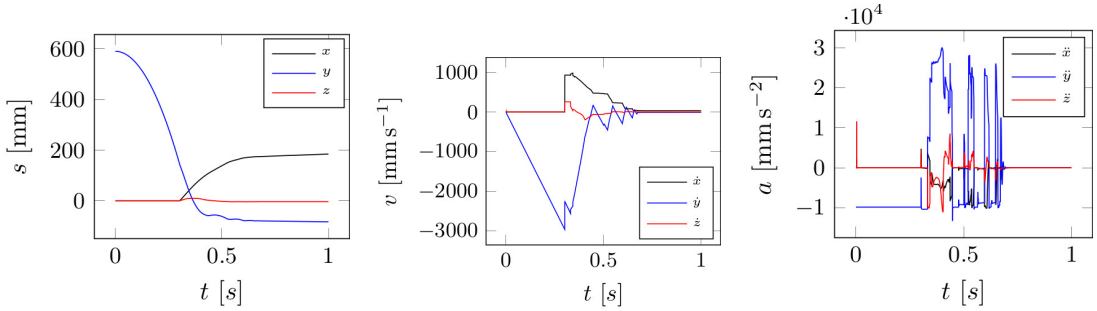


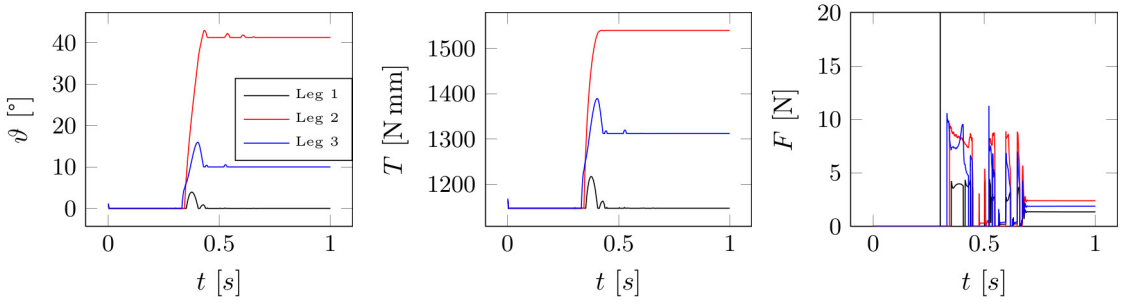
Figure 2.12: Extracted frames for the simulation of the soft-landing of the robotic lander MBS for the case of topple simulations

each pad and the terrain are all shown in Figure 2.14a–c, respectively. Finally, the trajectory of the lander's kinetic energy is lastly depicted in Figure 2.15.



(a) (b) (c)

Figure 2.13: Information about the state of the lander's MBS along the vertical axis: (**a**) the displacement of the center of mass; (b) the velocity; (c) the acceleration [176].



(a) (b) (c)

Figure 2.14: (**a**) The curves representing the angular rotation of each of the three leg structure revolute joint; (**b**) the curves representing the torque response of each of the three VRD's; (**c**) the curves representing the contact force between each of the legs with the soil [176].

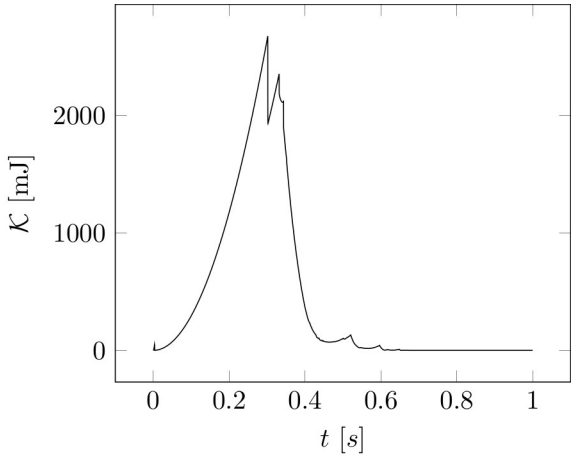


Figure 2.15: Kinetic energy evolution over time of the lander MBS [176].

2.2.5 Discussions

The outcomes of the soft-landing simulations on a flat surface and an inclined surface, which have been reported in Sec. 2.2.4, are examined in detail within this section. Additionally, some significant outcomes from the numerical solution of the theoretical-based model, presented in Sec. 2.2.2 and implemented in MATLAB, are provided in order to validate the presented methodology.

Comparison with the mathematical Model

For two separate cases involving the analytical-based simulation, comparisons between the ADAMS simulations and the theoretical model are conducted. For the first scenario, a robotic lander leg's length equal to l_c, which corresponds to the distance between the pad's closest contact point and the lander joint axis, is chosen. This parameter is considered to be conservative. The length for the second example is selected to be l_m, where this last variable is the mean distance between the contact point and the joint axis. This is done for the following reason: the pad in the ADAMS model, which is a rigid body with its own semicircular geometry, is different from the analytical model, which treats the pad as a point. Indeed, the shape of the pad causes the contact point to fluctuate along the pad profile over the whole contact process, thus changing continuously the lander's leg length. On the other hand, in the analytical model, it is always at a fixed distance from the lander joint axis. This discrepancy between the models has the drawback of continuously changing the lever arm during soft-landing, hence affecting the dynamics.

Comparisons are performed for the scenario of soft-landing on a horizontal surface, i.e., the scenario depicted as *Case 1.* Additionally, an extra simulation in ADAMS is performed excluding ratchets and is compared to the others in order to detect the impact the ratchets have on the lander's overall dynamics. For the two leg length values selected, Figure 2.16 aims at providing a comparison between ADAMS simulations and the MATLAB-based ones. The lander's acceleration, the force acting on the lander pad, and the leg-joint rotation are compared in the

top row, middle row, and bottom row, respectively. The curves in black represent ADAMS simulations including ratchets in the simulated environment (ADAMS-R), the blue curves indicate ADAMS simulations excluding ratchets from the simulated environment (ADAMS-NR), and lastly, the red curves represent simulations conducted exploiting the analytical model discussed in Sec. 2.2.2. It's worth noting that the figures only account for the lander's initial bounce. This is due to the fact that after the initial bounce, the lander behavior varies between simulations. In fact, the ratchet is not implemented by the analytical model. On the contrary side, this device exists in ADAMS and has a great impact on the system's behavior, especially after the latter stops decelerating and starts moving upwards.

Similarities between the trends and the predicted values can be discovered by comparing the curves. More specifically, the theoretical model predicts that the robotic lander will decelerate with $a \in [3.1, 4.5]g$ for a lander leg's length taken as l_c and $a \in [2.7, 4.0]g$ for a lander leg's length taken as l_m, on the other hand, simulations conducted in ADAMS simulation shows an acceleration of $a \in [3, 3.5]g$. The same can be said for the contact force and the rotation of the joint of the leg joint. In particular, for the former, a contact force $F_c \in [5.4, 10]$N has been found using the MATLAB theoretical-based simulation model, and the same range of force is also seen in the ADAMS simulation. Finally, for the latter, it can be shown that in both the simulations based on the theoretical model and the one using ADAMS, the highest values of rotation is slightly less than 30°. The ratchets' action, whose purpose is to restrict the rotation of the lander legs, i.e. allowing the leg to rotate in just one direction, is what provokes the discontinuities which are notable in the curves representing the ADAMS simulations. These discontinuities are visible in the acceleration and force graphs from Figure 2.16. On the other hand, those discontinuities are not present in the curves representing the analytical model, which does not incorporate the ratchet's action.

Moreover, it is possible to see that the theoretical model's leg length parameter, which has also been defined as the separation between the point of contact of the pad with the soil and the joint axis of the leg, has a significant impact on the curve's trend by contrasting the left and right columns of Figure 2.16's MATLAB-based simulations. In actuality, and it is particularly true for the contact force, the analytical curves suit the ADAMS curves better when l_m is chosen as the leg length. The analytical curve, however, has a higher peak at the start of the impact phase compared to the ADAMS ones. This is because, at that instant, the contact point is further away from l_m. Additionally, as the impact phase progresses and then the legs rotate, the contact point gets closer to the mean one, ending then this point reaching the minimum distance l_c.

The impact of the ratchets on the robotic lander's overall dynamics may be easily observed by contrasting the black and blue curves in Figure 2.16. Particularly, the existence of the ratchet may be seen in the acceleration and force graphs as minor humps in the curves. Instead, the main impact of the ratchet in the robotic lander's dynamics can be seen clearly from the angle graphs. More in detail, it can be seen that the joint of the leg presents a lower angle value, and this is due to the frictional forces which occur between the ratchet and the

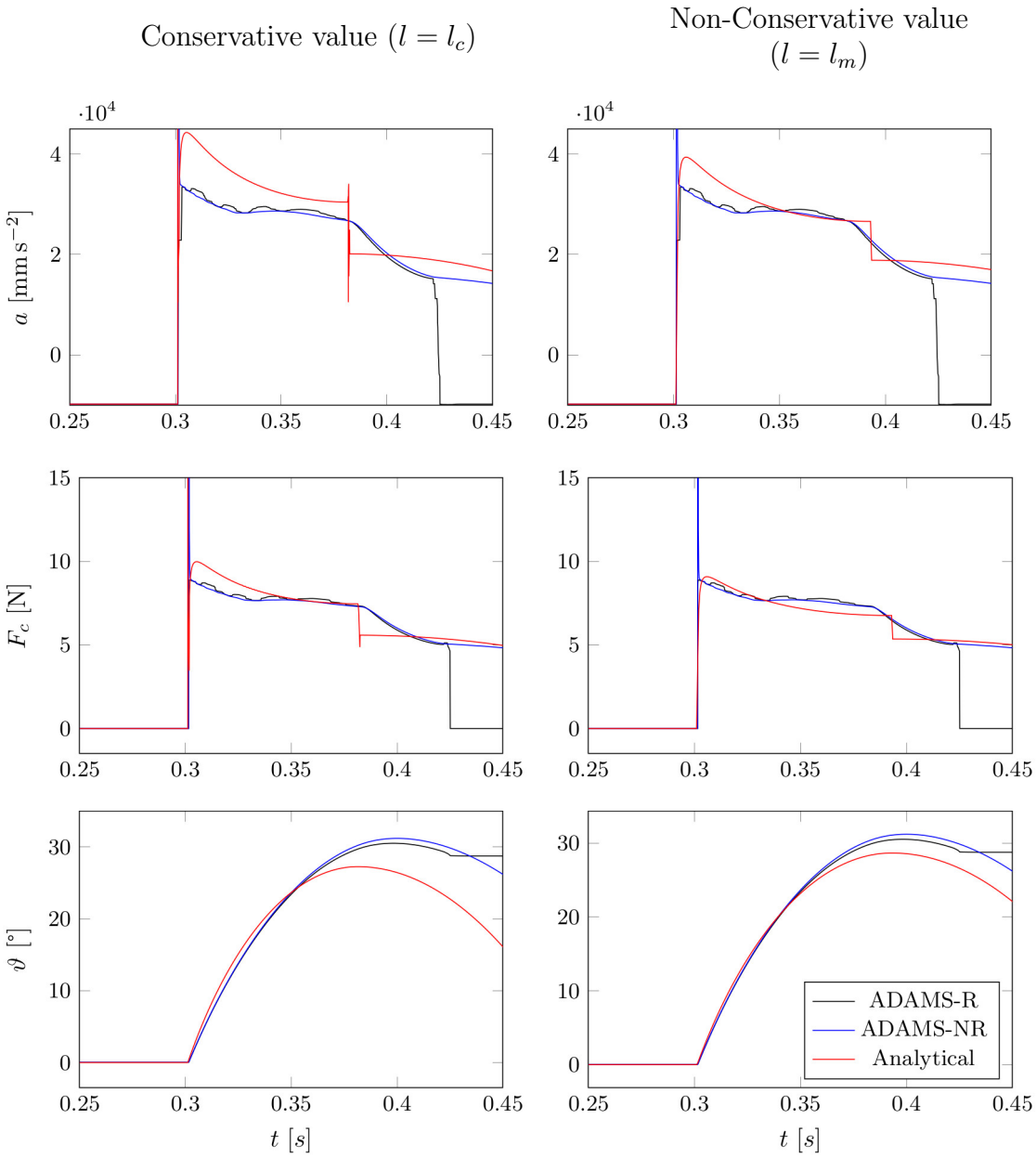
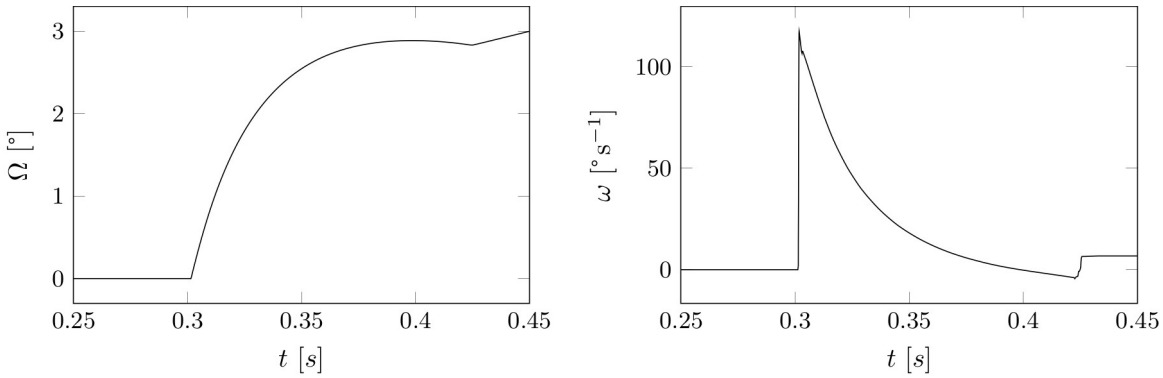


Figure 2.16: Comparison between ADAMS simulations with and without ratchets and the MATLAB-based analytical simulation for two leg length parameters chosen: in the top row the lander acceleration, in the middle row the contact force between soil and lander pad, and in the bottom layer the evolution of the leg joint rotation [176].

leg, thus generating a torque that opposes to the motion. This effect results in a lower angle for the leg joint. It is crucial to note that the ratchet has indeed been designed to readily deflect when goes in contact with the leg, therefore in this scenario, its influence is evident but only has a little impact on the robotic lander's overall dynamics. The dynamics of the robotic lander are affected more by stiffer ratchets, which can potentially result in the worst-case scenario of the legs never opening at all. It has been noted that exist limit values for the ratchet

elasticity, which has the major drawback of not deflecting enough to let the leg rotate, this leads to the leg locking and therefore subjecting the structure and the transported payload to extremely high vertical accelerations.

Another reason for the inconsistencies between the simulations conducted with the theoretical model and the ADAMS ones is that the theoretical model is assumed to be strictly bi-dimensional from the very beginning, and therefore can not capture phenomena strictly connected to three dimensions. Indeed, the robotic lander's chassis partially spins about its vertical axis during the soft-landing phase, which is permitted by the design of the lander legs. This fact translates into the effect that some of the robotic lander's energy is converted into rotational energy and is subsequently dissipated through lateral friction forces. Specifically, Figure 2.17 shows in a graphical way the just reported phenomena. In this figure, the robotic lander's rotation is shown on the left, and its velocity of rotation during soft-landing phase is shown on the right. It can be clearly seen at this time that the robotic lander's rotational energy has nearly totally dissipated in 0.15 seconds. Additionally, the contact model in ADAMS utilizes a nonlinear damper, on the other hand, a linear damper is utilized within the analytical model, which results in differences across the models. Differences arise as a result of this fact during the earliest phases of impact. The just linear damper actually produces stronger dampening forces at the point of contact. In particular, differences between the simulations conducted with the theoretical base model and the ADAMS ones are caused by the confluence of all the aforementioned elements. However, the findings of the two models exhibit significant conformity, confirming the approach despite the differences between the simulations conducted with the theoretical model and the ADAMS one, where the former is after all a two-dimensional idealization of a three-dimensional lander.



(a) (b)

Figure 2.17: (**a**) The lander structure center mass rotation around its vertical axis and (**b**) its angular speed around the same axis [176].

Case 1. Simulation on horizontal surface

Figure 2.8 shows how the robotic lander MBS's state changed during the course of the experiment. Specifically, Figure 2.8a,b,c) depict the vertical position; the

speed; and the acceleration of the center of mass of the robotic lander, respectively. These graphs show that the lander first touch the soil with a vertical speed equal to $v = 3.0\,\mathrm{m\,s^{-1}}$ at the simulation time equals to $t = 0.3\,\mathrm{s}$. Additionally, several abrupt variations in speed are seen, indicating that the robotic lander experiences bounces after contact. The dissipative forces specified in the contact problem cause the oscillation speed and amplitude to decrease gradually. From the same figure, it is evident that the lander experiences deceleration peaks with an amplitude of $|a| \in [3, 3.5]g$. It is worth noting that the acceleration's trajectory has been somewhat filtered to remove high-frequency deceleration peaks that occur at certain directly attributed to numerical instabilities. These may be ignored from a modeling standpoint because they have a minimal effect on the soft-landing "behavior" and are simply eliminated by using appropriate devices, which introduce damping, on the mounting points of the payload.

The lander's legs begin to rotate as they touch the ground because of the torque induced by the contact force generated during impact, and keep on doing so until the chassis stops decelerating. For the case of a soft-landing simulation on a horizontal surface, the lander's legs rotate equally up to the maximum angle ϑ, which has been found to be $\vartheta \approx 30^\circ$ for each leg. Note that the reactive torques produced by the VRD follow the same pattern. Figure 2.9a,b) shows these aspects, as well as the ratchets' effect. More specifically, the leg begins to rotate back to its original position after it has reached its maximum opening angle. However, the ratchet effect begins almost instantly; in fact, it engages and prevents this backward motion, thereby locking the joints of the leg, thus its orientation at the actual angle. The torque T, associated with this joint angle, has been found to be about $T \approx 1.5\,\mathrm{N\,m}$. The contact forces between the lander leg and the terrain are lastly represented in Figure 2.9c). As done previously for the acceleration, the contact forces have been somewhat filtered in order to eliminate the high-frequency spikes. The bounces of the robotic lander are visible in this image, and the contact force follows the same pattern as the joint rotation as would be predicted. Looking at the diagram once more, and comparing it with Figure 2.10b), which represents the kinetic energy of the lander MBS across the entire impact simulation, is, therefore, possible to conclude that the initial peak of the contact force, is the one responsible for dissipating the majority of the system's energy. The other spikes, on the other hand, are related to the lander's stabilization over the soil surface. Finally, the observed contact force's amplitude range is found to be $F \in [5.0, 8.7]\mathrm{N}$. Last but not least, Figure 2.10a) shows how the revolute joint in the ratchet structure rotates. The high peaks show the moments when the ratchet passes over the tooth on the leg's side and advances to the following tooth. In this instance, the ratchet surpasses four teeth. Instead, the lander's leg is attempting to turn counterclockwise, which is against the currently engaged ratchet, which is the cause of the low peaks that are noticeable.

Case 2. Topple simulation

The outcomes of a lander's soft-landing simulations over an inclined surface sample having a 1:5 inclination ratio are presented in the paragraph that follows.

In contrast to the previously described case, in this new case, the lander's state

is allowed to also vary along the two other axes, instead of the solely vertical one. The movement of the center of mass of the robotic lander, along the along x (black curve), y (blue curve), and z (red curve) is shown in Figure 2.13a). On the other hand, Figure 2.13b) shows the lander's speed again along the x,y and z axes. The acceleration components of the lander's center of mass along the 3 axes are shown in Figure 2.13c); and have been somewhat filtered in order to smooth and remove the high-frequency peaks, as done for the previous case. Although the vertical axis continues to be the primary axis of the motion of the lander, the other components in this circumstance can not be ignored. The lander impacts the underlying soil with a vertical speed equal to of v of around 3 meters per second, and the chassis experience acceleration peaks within a range of $|a| \in [2.1, 3]$ g. Finally, as seen in the previous case lander's bouncing behavior, before the stabilization, may be seen in the same figures.

The rotations of the revolute joints connecting the legs to the lander's chassis are shown in Figure 2.14a). It can be noted that the joints' rotation behave differently with respect to the previously described case. More precisely, the joint related to the upstream leg, which is the first leg to impact, "does not" rotate at all. On the contrary, the downstream legs experience the greatest rotation in the simulation, or ϑ approximately 42 degrees. Moreover, since the applied torque is a function of the relative joint rotation, this has influences on torques produced by the lander, as it can be noted in Figure 2.14b).

Additionally, the trajectories of the contact forces, exerted by the soil on each of the leg pads, are reported in Figure 2.14c). With reference to this figure, it can be seen that the upstream leg is the first which contacts the soil and assume temporarily a pivot behavior, hence converting part of the translational kinetic energy into rotational. Moreover, still with reference to the same figures, a high peak of force can be seen for the leg 1, which represents the moment it experiences contact with the soil. Figure 2.15 shows the peaks that are mainly responsible for the dissipation of kinetic energy; however, when the force peaks are moved forward in time, the rate of kinetic energy dissipation decreases. Finally, as it was easily predicted, these contact forces do not have the same modulus. Leg 1 has a lesser force since it is upstream and makes contact with the ground first; in reality, because its energy is converted from translational to rotational, less of it is transferred to the spring. More specifically, the observed force is $F \approx 5\,\mathrm{N}$, but for the two downstream legs, the force is larger, reaching values of $F \approx 10\,\mathrm{N}$.

2.3 Archimede rover

In this section, we aim to study and address an application belonging to the second macro group of mobile robotic systems for space applications, namely planetary rovers. In particular, this section aims to present the design and study of an innovative rover for space exploration, named Archimede. This rover is a four-wheel steerable vehicle that uses articulated legs as its suspension system. The leg joints are achieved by leveraging a mechanism called S-Structures, which allows the structure a certain degree of elasticity and lability, to provide some degree of impact resistance and ultimately higher operating speeds. The firmware and software of the rover are completely based on the ROS framework. In addition, this section will describe in detail each fundamental element of the rover, the subsystems that make up the rover, not only from a mechanical point of view but also from a software perspective. At the same time, this study aims to describe, model, and address the interaction with the ground that occurs while driving over terrain, while negotiating obstacles and during impacts. These aspects will later be validated in a simulated environment.

As with robotic landers, these systems are also primarily subject to interactions with the environment, and these occur through the wheels. It follows that this investigation also falls into the group of environment-robot interactions. Moreover, in this case, the interactions with the environment are not all undesirable phenomena. In fact, the interactions between the wheels and the ground allow the rover to move over it and to overcome obstacles. In addition, the rover, through its suspension system, is able to resist impacts and can therefore fall from a certain height. Furthermore, in this specific application, as will be seen below, contacts are constantly present even within the joints composed of the S-Structures, and these are the ones that provide the rover's suspension system with energy dissipation capabilities. Impacts also occur during normal driving and negotiation of obstacles. It follows that to design, simulate, and predict rover behavior, interactions and contacts must be properly modeled and addressed.

Moreover, another typical challenge affecting rovers, i.e. robot positioning, is addressed. More precisely, it will be outlined the development of the odometry sub-system, based on wheels encoder readings, and will be demonstrated its operation by comparing simulated Gazebo and experimental paths. Finally, an approach, based on the instantaneous center of rotation projection procedure, will be presented in order to provide a stable methodology to control a vehicle subjected to wheels steering constraints.

Summarizing, the contributions of this section can be listed in:

- The development of a novel four-wheel steering rover prototype for space exploration having articulated legs which grant it of a suspension system and high driving speeds;
- The development of the rover's ROS-based firmware and software;
- The development of a simplified dynamics model of the rover and the development of a realistic numerical model of the rover in ADAMS;

- The experimental validation of the proposed numerical models;
- The development of an odometry sub-system for the rover, its numerical and experimental demonstration, accuracy assessment, and development of a joint limits avoidance strategy based on the projection of the ICR and Ackermann steering;
- The creation of ROS packages for high-level control of the rover, and the modeling of the Gazebo simulation environment.

2.3.1 Introduction

Rovers are likely the most adaptable and useful machines in the domain of planetary exploration space missions. Within this field, these robotic systems have recently been utilized to control sensors and equipment [179], examine soil samples [180], and assess the harsh characteristics of the environments in which they operate [181, 182]. Rovers are being developed for manipulating tiny modules [183], moving sensors [184–186], and for transporting and carrying soil samples [187]. A sample of Martian soil is intended to be returned to Earth in the 2030s via the Mars Sample Return (MSR) mission concept, which is being directed by NASA and ESA [188]. It is predicted that a complex infrastructure composed of planetary rovers, landers, and tiny martian lifters will have to be necessary to accomplish this [189]. The Tianwen-1 mission's Zhurong rover, which was launched by the Chinese space agency in 2021, was delivered in the Utopia Planitia region of Mars [190]. A sample-return mission that is anticipated for 2030 is seen as a prelude to this expedition. Several different systems must be built to actively cooperate on the surface of the planet for these challenging sample-return missions, for instance, a robotic lander together with a sample-collecting planetary rover and an ascending vehicle [191]. According to the current NASA-ESA design for MSR, Perseverance, the Mars 2020 rover, would gather samples, and a second Sample Fetch Rover (SFR), constructed by ESA, will carry those samples to a two-stage Mars Ascent Vehicle (MAV), designed by NASA [189]. With respect to the same reference of Muirhead et al., it is vital to note that the planned traverse distance of 20 km in the allotted 150 sols would necessitate a substantially greater traverse rate than any of those typically provided by existing operating Mars rovers. This element, along with several other similar multi-agent approaches [183, 184, 192, 193], seems to demand speedier and, generally, more efficient rovers that can cover large distances in a shorter amount of time [194]. Indeed, all the planetary rovers that have been used in real missions have very low driving speeds. As an overview of the systems which have been actually used in real missions, the Perseverance planetary rover does have a maximum speed of nearly $4.2 \times 10^{-2} \mathrm{m\,s^{-1}}$, the NASA's MERs Spirit and Opportunity have a maximum speed of nearly $5 \times 10^{-2} \mathrm{m\,s^{-1}}$, Curiosity, on the other hand, has a maximum speed of nearly $4 \times 10^{-2} \mathrm{m\,s^{-1}}$ [179]; the Chinese rover Zhurong ($3.3 \times 10^{-2} \mathrm{m\,s^{-1}}$), while the Soviet Lunokhod 1 and Lunokhod 2, were designed in order to have two different operating speeds, respectively 0.28 and 0.56 $\mathrm{m\,s^{-1}}$. Additionally, rovers must be able to endure collisions throughout the whole space

exploration operations, most of which normally happens during the deployment phase of the system. However, collisions can arise also when negotiating and climbing obstacles and driving over a rough terrain [179]. These happen to be more severe as the driving speed increases. Within this context, it is proposed and presented an in-depth examination of the rover "Archimede", which is based on a novel suspension system known also as S-structure (named because of the S-shaped like force-deformation relation) [195,196]. The mechanism introduces a non-linear elastic response, which has the benefit to allow the rover to drive at higher speeds ($1-2\mathrm{m\,s^{-1}}$), as well as providing the vehicle with shock absorption capability and scientific payload stability. Similar planetary rover prototypes that have been found in literature, which are designed for high speeds operations as well, are: the "Lightweight Rover Unit" (LRU) from DLR [57], which has been intended to drive with a top speed of $1.1\mathrm{m\,s^{-1}}$; the NASA's K10 planetary rover, which is designed to have a top speed of roughly $1\,\mathrm{m\,s^{-1}}$; while the Dune rover [197,198], with a claimed maximum speed of almost $1.5\mathrm{m\,s^{-1}}$.

In order to fully understand how the robotic systems behave during real planetary missions, which include the involvement of planetary rovers, it is crucial to capture every aspect of their behavior and predict the hypothetical scenarios which they may encounter. The challenge of simulating the kinematics and the dynamics of the rover, which must be deployed, is essential and directly tied to the mission's overall success. It follows that it's highly important to anticipate every scenario that the rover could run across. In contrast to the traditional methodologies, Kane's method makes it easier to derive the dynamics of complicated systems [199,200]. As an instance, Hussein et al. used Kane's method in order to derive the EOMs for a three-link planar arm [201].

With promising results, Lindemann demonstrated the performance of the Rover dynamics by numerical simulations utilizing the commercial software package MSC ADAMS [202]. The same software package has been used in another implementation that has been demonstrated by Benamar et al. [203]. By employing a pseudo-coordinates model, Chen and Genta demonstrated how the dynamics of rovers, used for space exploration purposes, can be characterized without taking into account the mechanics of soil [204]. Other researchers have made such claims as well, such as [205]. On the contrary, a large number of researchers have investigated how wheels interact with underlying soft terrains, frequently by simulating the soil using models from the field of terramechanics [39,206]. For instance, by using the Bekker terramechanics theory [207], Yang et al. constructed a discrete-element model for lunar rovers [208]. Moreover, in the research of Ishigami et al. a similar implementation is provided [209]. The commercial software package Simpack has been used by Schafer et al. to simulate rovers and soft terrain [116,210–212].

Other research investigated the interaction that occurs between a wheel and soil in a vacuum [213]; Additionally, Chen et al. investigated the kinematics, dynamics, and the control of the trajectory of a rover on soft soil [214].

Within this study, it is taken into consideration a wheel-soil interaction model that can be applied in the context of contacts in MBS and is defined by a massless spring-damper system. The study presented in this section is based on the

methodology given by Chen and Genta [204]. There are several other contact models (penalty-based ones) for MBS dynamics, which find the basis in the purely elastic Hertz contact theory, that may be found in the literature. The model from Lankarani and Nikravesh [215] and the one from Hunt and Crossley [216] are the two most significant and popular models. In this study, it has been chosen to utilize a Kelvin-Voight model, i.e. a visco-elastic formulation that describes the normal contact force with a non-linear spring-damper system [217].

For what concerns the development of the analytical model of the rover, which will be derived in the next paragraphs, it is very important to point out first that the main objective is to focus on and characterize the "Archimede" rover's overall behavior, rather than concentrating on the specifics of the of the wheel-soil interaction and trying to characterize the phenomena occurring at this specific phase. Secondly, the usage of mechanical elements such as the "S-Structures" [218] within the Archimedes rover makes it a very complex and very difficult system to model in a mathematical and formal way. In fact, as will be seen below, the rover is composed of 31 DOFs, and between one module and another of the S-Structures there is a whole series of internal contacts that significantly impact the overall dynamics [195, 196]. The wheel-soil interactions are reduced to a simple non-linear spring and damper in a parallel system with dry friction [217], while a rigid MBS is taken into account as an early study. Moreover, in order to demonstrate how Kane's method can be utilized to generate an analytical model for the complicated dynamics of the Archimede rover, it has been elected to implement a lumped-element model for the joints in the leg. Several Runge-Kutta RK4 solver-based numerical calculations of this model have been conducted, and the results are compared with both an experimental prototype and a fully modeled rigid MBS simulated in MSC ADAMS. Utilizing high frame rate photography, data had been collected. The results demonstrate good matching between the three models, particularly between the experimental and semi-analytical methods, despite the numerous sources of error. Finally, we give a thorough explanation of the sources of uncertainty and how the parameters are characterized.

This section is therefore structured as follows: in Sec. 2.3.2 are reported the design of the Archimede rover prototype with a description of every fundamental element; moreover, it is described the electronic setup of the rover together with a description of the software. In Sec. 2.3.3 is presented the detailed modeling of the Archimede rover. In Sec. 2.3.4 is reported the process for the derivation of the dynamics model of the rover with Kane's method and the definition of the semi-analytical model; moreover, it will be outlined the contact model that has been used together with a brief overview of how the terrains are created. In Sec. 2.3.5 is reported first the description of a simplified direct differential kinematics model, that will form the basis for the odometry sub-system; secondly, it will be presented the description of an inverse differential kinematics leveraging the general Ackermann steering, and that will form the basis for the control of the rover. In Sec. 2.3.6 is presented an approach that first leverages the Ackermann general steering, and subsequently, it projects the computed ICR into its admissible surfaces, in order to take into account lower and upper joints limits of the rover. In Sec. 2.3.7 is shown the modeling process of the complete model

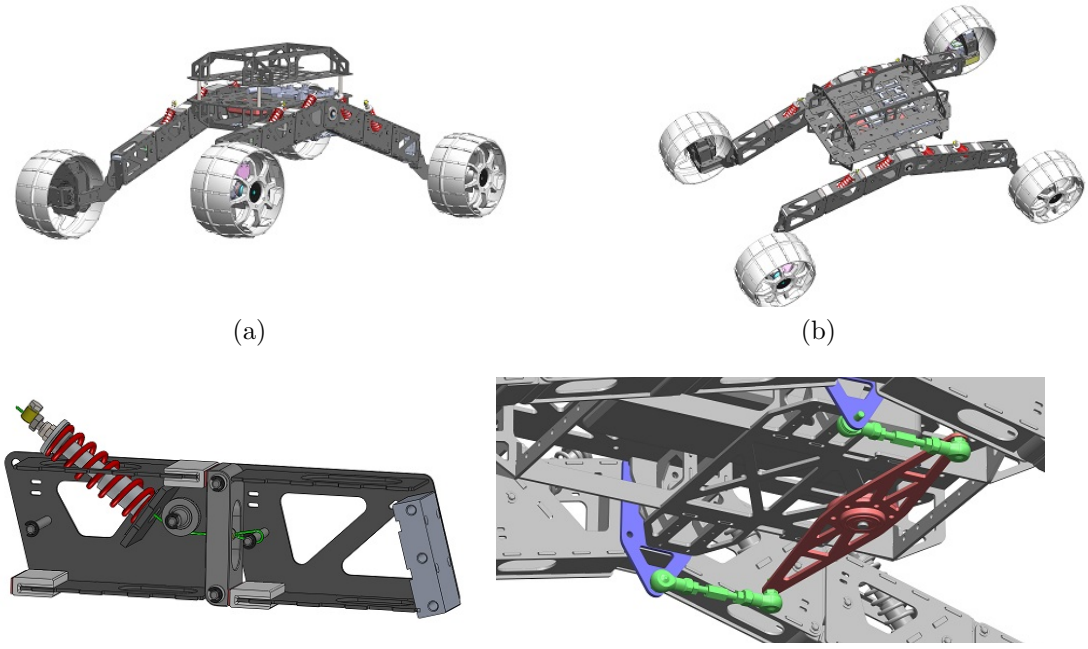
of the Archimede rover in the ADAMS software package, while in Sec. 2.3.8 is described the ROS-Gazebo setup and modeling process together with the ROS packages developed and needed. In Sec. 2.3.9 are reported the results obtained for the demonstration of the odometry sub-system operation. In Sec. 2.3.10 the validation process of the semi-analytical dynamics model, through direct comparison with the ADAMS model and the experimental prototype, for two different test case scenarios, is presented. The results of this comparison are reported as well within this paragraph. Finally, a sensitivity analysis, of the semi-analytical model's key hyperparameters is performed and presented, in order to characterize their influence. In Sec. 2.3.11 the results of the validation of the dynamic models are discussed with emphasis on the comparison between the different methodologies.

2.3.2 Prototype description

The basic design of the presented "Archimede" planetary rover prototype is based on the ideas described in [195] and [196] by Seriani et al. In further detail, the rover is a four-wheel steering rover with independent wheels that are each attached to the chassis of the rover through an articulated and flexible leg. The CAD model of the rover can be seen from different orientations in Figure 2.18a and Figure 2.18b. More precisely, each leg of the Archimede rover is made up of three consecutive modular bodies (leg's modules), in which the pair of adjacent modules is connected by a thin link. This pair of adjacent modules' rigidity is guaranteed by a preloaded linear spring. The just briefly described mechanism is known as the S-Structure mechanism, and Figure 2.18c shows a section view of it. As a result, the two leg's modules are related by a complicated elastic joint, which is first thoroughly detailed in [195].

In accordance with Figure 2.18, the Archimede rover is constructed to have a lever mechanism that connects the two leg bogies. This component has the main role of acting as a motion coupler in between the bogies, as illustrated in Figure 2.18d. To put it more specifically, the bogies are restrained from rotating in the same direction at the same time. Moreover, it has the effect of maintaining the rover's chassis with an angle midway with respect to the bogies' orientation.

Figure 2.19, on the other hand, shows how the Archimede rover's wheel hub is constructed and its components. As can be seen in the figure the wheel hub is constructed in a way such that there is no offset between the wheels' contact point and the steering axis. Moreover, the driving and steering axes are incident and orthogonal. Thus simplifying the rover's kinematics. Two separate Dynamixel XM430-350RW electric motors are installed in the wheel hub; one controls the steering joint, while the other the wheel rotation. It must be reported that with this configuration, the physical dimensions of the wheel combined with the ones of the adjacent articulated leg introduce lower and upper constraints to the steering joint. These are referred to as physical joint end-stops inherent to the steering joint, which have been denoted with $\delta_{j,min}$ and $\delta_{j,max}$ for the generic leg j. The just described physical steering joint end-stop is shown in the second row of Figure 2.19. The joint limits will be taken into account via software and it will



(a) (b)

(c) (d)

Figure 2.18: In (a) and (b) different views of the CAD assembly of the Archimede rover; in (c) a section view of the single S-Structure is shown in which we see: the two contiguous modules, the connecting link, the linear preloaded spring, and the redirecting pulley; finally, in (d) a view is shown of the rover, showing the lever mechanism which constrains the two bogies to rotate in opposite directions [219].

be seen how in the next paragraphs. Finally, it is worth pointing out that this arrangement attempts to simplify the wheel-hub system.

Every component that has been previously mentioned and included in the actual Archimede rover prototype, with special attention paid to the design and manufacturing decisions, is shown in Figure 2.20. It can be seen that sheets of carbon fiber composite (CFC) are used and joined together with glue or fastened in order to build the rover's chassis. On the other hand, Four CFC plates make up the enclosure for both the preloaded spring and the deviation pulley, which are located inside the single leg's module. This leg's module can be better seen in Figure 2.20b. On the other hand, three-dimensionally printed Polylactic Acid (PLA) polymer is used to create the connecting link between the two adjacent modules.

On the other hand, for what concerns the joints of the rover Archimede, these are placed in various parts of the rover. First, in the leg's modules can be found steel axles sustained by bearings made of low-friction polymer which make up the revolute joints between the adjacent modules and the interconnecting link posed in between them. Subsequently, each leg of the rover is joined to the main body of the rover by means of a revolute joint, which can be seen in Figure 2.20c and Figure 2.20d. This connection is made up of a steel shaft that is fastened to the

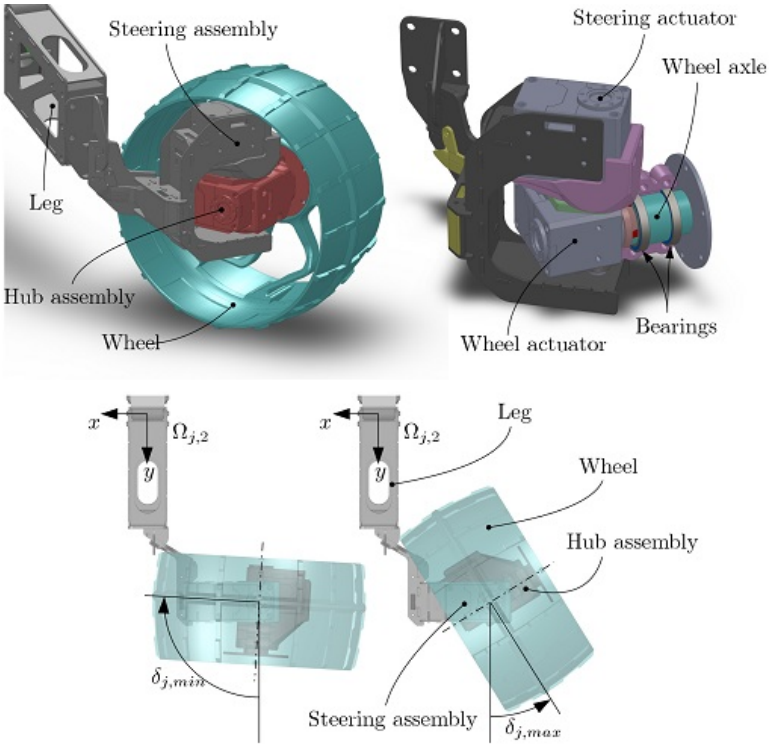


Figure 2.19: Mosaic showing the wheel assembly: in the top, the main components of the wheel hub are shown; in the bottom, the steering angle joint limits $\delta_{j,min}$ and $\delta_{j,max}$ are indicated [219].

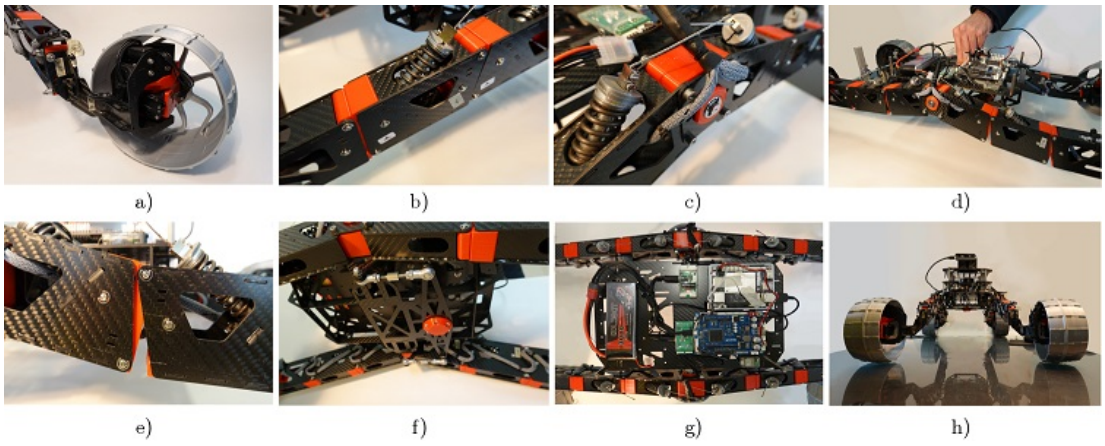


Figure 2.20: In (a) a three-quarter view of a wheel hub; in (b) a view of the single module of the leg; in (c) a view of the revolute joint connecting the leg to the rover chassis; in (d) the movement of the articulated legs under an applied vertical load; in (e) particular view of the movement of two contiguous modules, in which the small connecting link can be glimpsed, under the action of a vertical load; in (f) the view from the bottom of the lever mechanism; in (g) a view from the top of the electrical and electronic components and wiring; in (h) a view from the front showing the assembled rover [219].

chassis of the rover and is sustained by ball bearings. Subsequently, a long CFC fabricated arm with two spherical joints one at each end is hinged at the center of the rover's belly; then to each of the extremities of this arm, a steel connecting part is connected to the arm and the adjacent rover's leg. The joint between the steel connecting body and the adjacent leg is of type ball; and the overall described system makes up the lever mechanism underneath the rover, which can be better seen in Figure 2.20f. Finally, as already seen, in each of the wheel's hubs are present two revolute joints. One provides steering functionality to the wheel, the other one allows the wheel to rotate around its axis, i.e. it provides the driving functionality to the wheel.

Finally, the chassis of the Archimede rover is composed of three layers that are separated from one another by steel spacers, as shown in Figure 2.20(g–h). The "power and electronic heart" is located in the first layer and includes the lithium batteries pack, the Single Board Computer (SBC), the Robotis OpenCR ARM Cortex-M7 based control board, and the power distribution system (PDS), which will be better described in the next paragraph. The two-dimensional Robotis LDS-01 lidar scanner is located on the second level, while the top level is reserved for the stereo-camera.

Electronics setup and Power Distribution System

As described above the first level of the chassis of the Archimede rover is dedicated to the electronics, the batteries pack, and the power distribution system. The core and the brain of the Archimede rover can be thought to be composed of two distinct boards: the first is a Lattepanda v1 board, which is a Single Board Computer (SBC), and has installed Ubuntu 18.04 as the operating system, ROS, and related software, and it's responsible for taking care of most of the high-level operations and managing the high-level software, such as the main ROS packages. The second board instead is an OpenCR 1.0 (Open-source Control module for ROS), and it's used for mid-level control operations. This board is connected to the SBC through a USB serial interface which also provides the needed power supply to the board. This board is responsible for acquiring measured data from sensors, communicating with the SBC, and most important reading and commanding the eight motors through the RS485 serial interface.

With the only purpose of do not overload the boards too much, it has been decided instead of plugging all the motors into the OpenCR board, to split the bus of the motors and create two distinct circuits. One exclusively for the data bus, and one only for the power supply needed to run the motors. The boards are powered by a separate power source, which is separated from the motors' one. More precisely, from the OpenCR board comes just two data lines, which they go respectively on the left and right side of the rover. Taking as example the right line, this is responsible to provide data to the four motors belonging to the right side of the rover. This line is then merged with the power lines, and subsequently, the whole bus is split into two lines, where they go respectively to the front and the back wheels of the right side of the rover. The same is mirrored for the left side of the rover. From the description above, it can be assumed that the topology of the data bus follows a tree topology. Finally, the power source used for the

power lines of the motors is a LI-PO battery of 11.1V, 3000mA 19.98Wh, while the power source used for the two control boards is a portable power-bank.

Software

The rover makes use of the ROS framework and custom Python and C++ packages, or third-party released ones.

The core software of the rover is divided into two parts: ROS packages on the SBC and firmware on the OpenCR.

Within the ROS network, in which the prototype of the rover works, the usually elected configuration is the one in which the rover functions as the slave and the remote computer is designed to be the master in this master-slave configuration. With this design, the SBC can merely execute the fundamental components while a remote computer within the ROS network does all the computationally heavy operations. The main package of the rover's software, which implements planning, high-level control, as well as teleoperations through remote input devices; the visualization package; and the post-processing package are the packages that are installed and run on the master machine. On the other hand, in the slave machine it can be found: the package which implements the rover's kinematics, which expects the high-level control inputs from the master machine (or a connected input device), subsequently it computes the inverse kinematics, and finally provides commands to wheel actuators; the main package, which includes the core functionalities such as the driver for the motors, position estimation through wheels odometry, and provides internal state sensing. The bi-dimensional Lidar scanner sensor and the RealSense D435i stereo-camera are handled by other small and dedicated packages installed on the SBC. Finally, Figure 2.21 provides a schematic illustration of the software architecture and the ROS network. Some of the ROS packages used will be better seen in the next paragraphs.

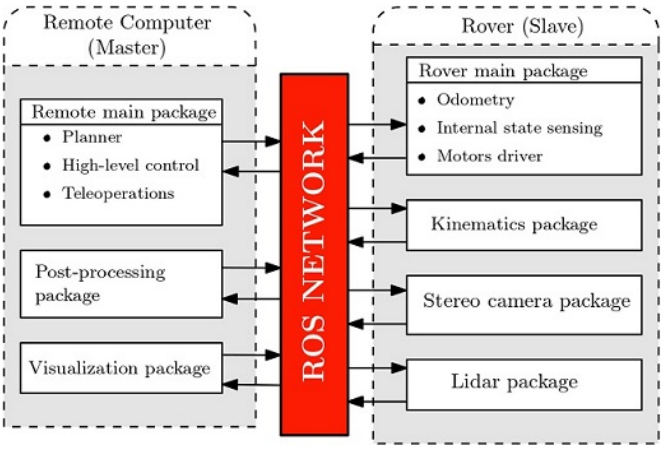


Figure 2.21: Schematics of the rover's software architecture [219].

As previously stated, the robot's kinematics package enables the high-level control variables, provided by the user or for example by the navigation stack algorithm, to be broken down into wheel states actuators commands. These are the angular position the steer motors should assume and the angular velocity for the

driving wheels; subsequently, these commands are communicated to the driver of the motors, that first translates the received commands and then it applies them, through serial communication, to the individual motors. The kinematics package also allows the rover to drive in a variety of other ways, such as symmetric Ackermann steering, car-like steering, parallel drive, lateral drive, general Ackermann steering, pure in-place rotation, etc. The same functionality handles the wheel hub's lower and upper joint constraints. All the just mentioned notions will be seen more in-depth in the next paragraphs.

2.3.3 Modeling of the Archimede rover

The aim of this paragraph is to provide the kinematics and the basis for the dynamics model of the Archimede rover prototype.

In this vehicle, the main challenges come from the S-Structure model, since this mechanism is known to be very complex due to internal contacts.

It has been seen that in principle the exact analytical model for a single S-Structure (two contiguous leg's modules, the spring, and the connecting rod) can be easily derived and the contacts handled and addressed by managing the dynamics of the system with a Finite State Automaton (FSA). In this FSA are identified four different states: S_1 i.e. no contacts at all between the two bodies (the dynamics is equivalent to the one of a three-link planar arm), S_2 i.e. contact occurs only at the bottom of the modules, S_3 i.e. contact occurs only at the top of the modules, S_4 i.e. contact occurs both the bottom and the top of the modules. A graphical representation of the identified states is reported in Figure 2.22, while the FSA schematic representation together with the state transition functions is reported in Figure 2.23a. An example result of this approach is reported instead in Figure 2.23b. Finally, the Simulink model workflow used, taken as an example for the dynamics simulations of the single S-Structure module, is reported in Figure 2.24. In this specific case, the first module is hinged with a prismatic joint, allowing only vertical motion.

However, when the number of consecutive S-Structures grows, the number of internal contacts that need to be managed grows as well. It follows that the complexity of the analytical model grows exponentially and becomes extremely challenging to model such a system in a purely mathematical, formal, and exact way.

As claimed in the previous research by Seriani et al. in [195] a revolute joint with an applied equivalent preloaded torsional spring can be used in place of each S-Structure joint in order to simplify it. This crucial assumption will serve as the foundation for the simplifying assumptions which will be introduced to the dynamic model.

Therefore, three main assumptions are then stressed in order to derive the simplified analytical model for the dynamics of the Archimede rover: (i) the bodies constituting the rover are handled as perfectly rigid bodies; (ii) the revolute joints are considered to be ideal, indicating that there are no joint deformations or joint dissipative forces; (iii) the complicated joint present inside the S-Structure is reduced to a single revolute joint with a lumped-parameters and an appropriate

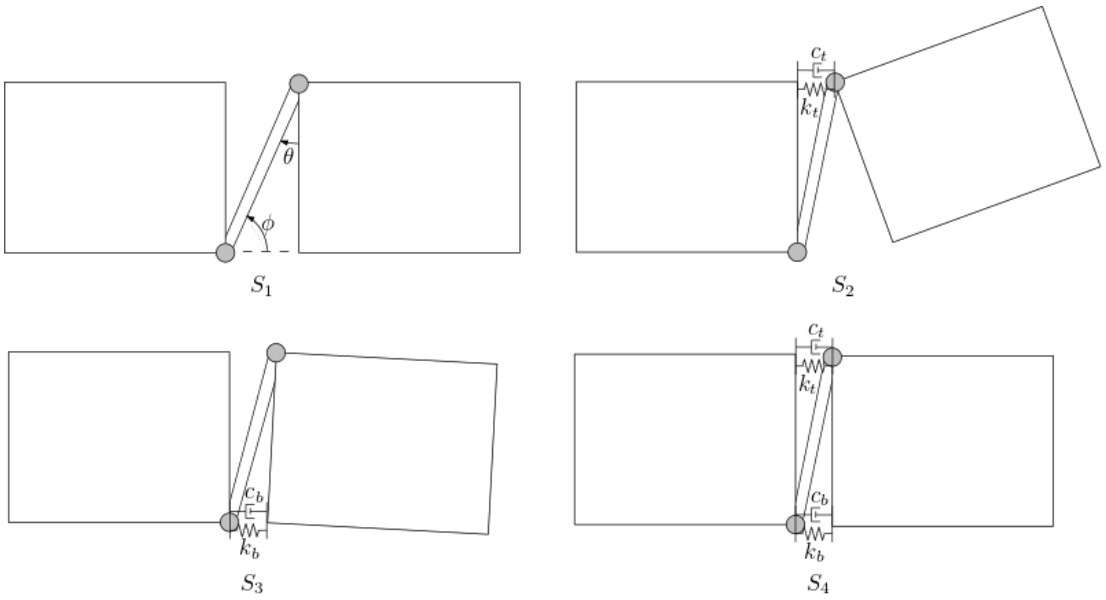


Figure 2.22: Schematic diagram showing the identified states the S-Structure mechanism assumes.

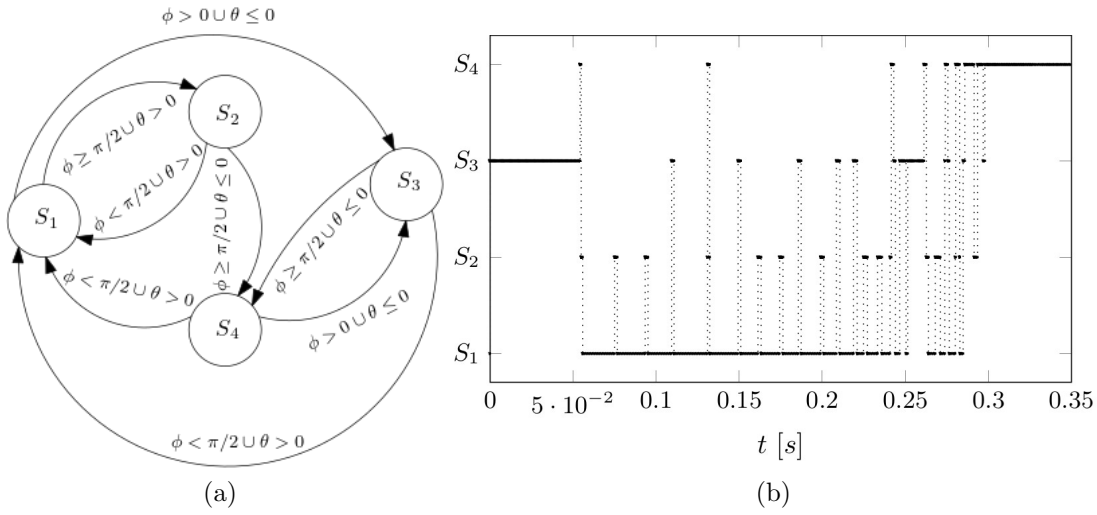


Figure 2.23: In (a) the contact FSA diagram showing the four states and the transition functions; in (b) the active state assumed by the FSA during a sample simulation with focus on the first 0.35 s

applied non-linear torsional spring.

With reference to Figure 2.25a, consider an inertial reference frame $(\mathbf{O}, \bar{\mathbf{e}}_O)$ that is fixed in the three-dimensional Cartesian space having origin $\mathbf{O}$ and $\bar{\mathbf{e}}_O$ represent the basis vector having components $\hat{\mathbf{e}}_{O_x}$, $\hat{\mathbf{e}}_{O_y}$, and $\hat{\mathbf{e}}_{O_z}$. Consider then, a second reference frame $(\mathbf{G}, \bar{\mathbf{e}}_G)$, which is fixed to the rover's body and is situated in its center of mass (COM) $\mathbf{G}$. Moreover, the two frames are rotated in reference to one another according to the *RPY* (Roll-Pitch-Yaw) convention.

Subsequently, consider the reference frame $(\mathbf{B}_r, \bar{\mathbf{e}}_{B_r})$ which is fixed to the right

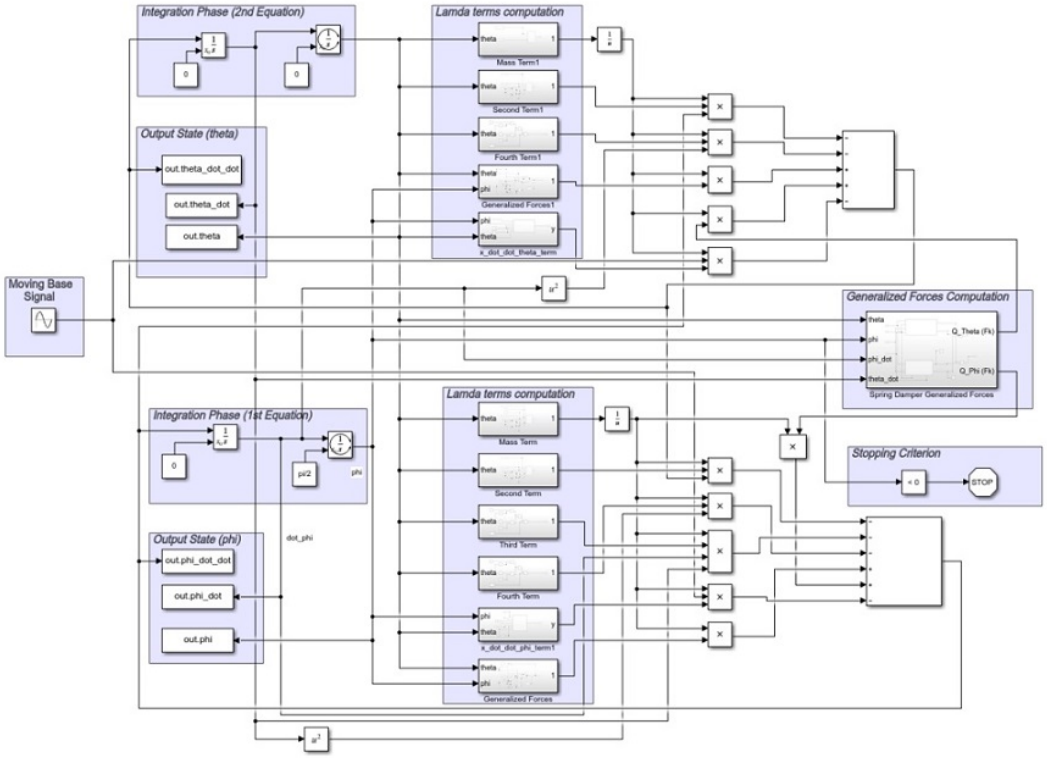


Figure 2.24: Simulink workflow model used for the dynamic simulations of the single S-Structure module.

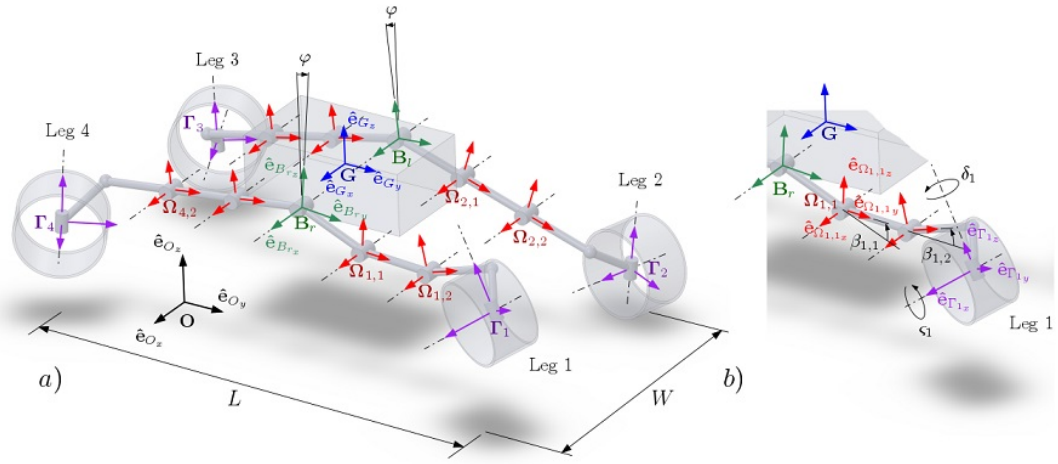


Figure 2.25: In (a) the schematic representation of the simplified model of the rover is shown, with special care on indicating the involved reference frames; in (b) an expanded view can be seen of the first leg, showing its generalized coordinates [219].

bogie of the Archimede rover. The just introduced reference frame is allowed to rotate freely by an angle ϕ around the x axis of the previously defined frame $(\mathbf{G}, \bar{\mathbf{e}}_G)$, i.e. the rover's one. For what concerns the left bogie which is defined by the reference frame indicated with $(\mathbf{B}_l, \bar{\mathbf{e}}_{B_l})$, instead of being independent, it

is imposed it to be coupled to the one describing the right bogie, rotating by the same amount but in the reverse direction, i.e. $-\phi$.

Again with particular attention to Figure 2.25(a-b), to the generic i-th module constituting each j-th leg of the rover is assigned a fixed reference frame $(\mathbf{\Omega}_{j,i}, \bar{\mathbf{e}}_{\Omega_{j,i}})$, where $j = 1, \ldots, 4$. Referring still to the j-th leg, the reference frame fixed to the i-th module is left free to rotate with an angle $\beta_{j,i}$ along the x axis of the upstream reference frame. On the other hand, referring now to to Figure 2.25b), the reference frame $(\mathbf{\Gamma}_j, \bar{\mathbf{e}}_{\Gamma_j})$ describes the wheel-hub of the j-th leg. The letters L and W, which stand for the wheelbase and track dimensions, respectively, represent the general properties of the considered wheeled system. Finally, for what concerns the wheel-hub of the rover, referring still to the j-th leg then the j-th steering and driving angles are described by the angles δ_j and ζ_j, respectively.

Having shown the rover's simplified model, the reference frames introduced and described how they are related to each other, now the attention will be posed on the DOFs which are needed to fully describe the system. Indeed, a total of 23 DOFs describe the rover's simplified model in three dimensions of Cartesian space. Moreover, by designating $\boldsymbol{q}$ as the vector of the system's generalized coordinates, it is evident that $\boldsymbol{q} = [x, y, z, \theta, \chi, \psi, \phi, \{\beta_{j,i}\}, \{\delta_j\}, \{\zeta_j\}]$. The components of the vector $\boldsymbol{q}$ are x, y, and z which are the rover's position vector $\boldsymbol{p_r}$ components in the space and relative to the inertial frame; θ, χ, and ψ which represents the rover's respective yaw, pitch, and roll angles, respectively; $\{\beta_{j,i}\}$ which represent the vector of the modules' relative rotation; $\{\delta_j\}$ and $\{\zeta_j\}$ which represent the steering and driving angles' vectors, respectively. Finally, it must be specified that $i = 1, \ldots, 2$; and $j = 1, \ldots, 4$.

Finally must be defined and assigned the geometric and physical properties of the bodies composing the simplified rover model. These are inferred from the properties of the real Archimede rover prototype. More precisely, these bodies making up the rover's MBS are represented using lumped parameters. The rover's chassis is simply described as a box and has a complete inertia tensor $\boldsymbol{I_R} \in \mathbb{R}^{3\times3}$ as well as mass m_R. The arm inclination Δ, the length l_{SA}, and the offset d_{SA} along the x-axis of the rover reference frame are the geometric characteristics defining the bogies. Additionally, they posses physical characteristics of mass m_{SA} and a complete inertia tensor $\boldsymbol{I_{SA}} \in \mathbb{R}^{3\times3}$. Consider again the generic j-th leg in accordance to Figure 2.25b). The properties of its generic i-th module are the length $l_{j,i}$, the mass $m_{j,i}$, and a complete inertia tensor $\boldsymbol{I}_{j,i} \in \mathbb{R}^{3\times3}$. In general, the COM $\mathbf{G}_{j,i}$, does not match the geometric center of the module. The three offsets along the x, y, and z axes, which are all defined and measured in the terminal module of the j-th leg reference frame, define the wheels COM, and are defined as d_{j,w_x}, d_{j,w_y} and d_{j,w_z} respectively. The mass $m_{j,w}$ and a complete inertia tensor $\boldsymbol{I_{j,w}} \in \mathbb{R}^{3\times3}$ of the wheels represent their physical characteristics, while the radius $R_{j,w}$ it's geometrical ones. The single bodies, even though they belong to distinct legs, share the same physical and geometric properties for symmetry considerations and for the stability of the overall system.

The models for the kinematics as well as the dynamics for the aforementioned simplified and approximated model of the rover Archimede have been further

developed in Python in an exact form by using Kane's approach[1].

As stated above the simplified model is based on the assumption of simplifying the generic S-Structure complex joint with a simple revolute joint in which is applied a non-linear torsional spring. This assumption then translates into the fact that in the simplified model a preloaded non-linear torsional spring is operating on the revolute joint $\tau_{j,i}$, which connects the generic modules i and $i+1$, for each leg j and ensures the stiffness between the two bodies. Considering that the preloaded spring's reaction, while operating in between the two leg's modules, is a function that presents discontinuities around the zero, its response may be approximated, as claimed by Seriani et al. in [195], by the following function,

$$\boldsymbol{M}_{k,\tau_{j,i}} = -\left(\frac{2M_{0,\tau_{j,i}}}{\pi}\right)\arctan\left(f\beta_{\tau_{j,i}}\right) - k\beta_{\tau_{j,i}}, \tag{2.13}$$

where k, $M_{0,\tau_{j,i}}$, and f are the equivalent stiffness of the torsional spring, the preload torque the joint $\tau_{j,i}$ is subjected, and an approximation factor regulating the harshness of the step response function, respectively. The relative rotation of the joint $\tau_{j,i}$ is represented by the parameter $\beta_{\tau_{j,i}}$. The general response function of a torsional spring including preload is illustrated in Figure 2.26 along with different approximating functions, where the sharpening factor f has been modified. Finally, by using this approach, it is possible to consider the sophisticated S-Structure mechanism, which after all contains complex internal contacts to manage, as a significantly more manageable revolute joint with an applied non-linear torsional spring. Additionally, the model's simplification translates into a decrease in the complexity which results in a loss of eight DOFs. The viscous torsional dampers, which are represented by the coefficients $c_{\tau_{j,i}}$ and have been modeled alongside the elastic spring torques, shall hereafter be referred to as c_q since they are all the same. These have been introduced in order to provide the system with the damping effect that has been observed in the real S-Structure, an intricate phenomenon that is probably caused by internal collisions and frictional forces and torques [195].

2.3.4 Dynamics model

Kane's method is used to calculate the EOMs for the entire MBS and is described in this paragraph. In order to derive the EOMs for complicated MBS, Kane [199] developed a methodology, which now is called after his name. In particular, Kane's method enables quicker calculation of the EOMs with respect to the traditional Euler-Lagrange approach. Both the generalized inertia forces F^* as well as the generalized active forces F are evaluated within the approach. The D'Alambert principle dictates that the sum of these generalized forces must be zero. Given a complex MBS with n DOFs and N_b bodies; the equations for Kane's dynamics are defined as follows:

$$F_r + F_r^* = 0, \qquad r = 1, \ldots, n \tag{2.14}$$

[1]The data and the source code are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/rover-multibody-simulator.git`.

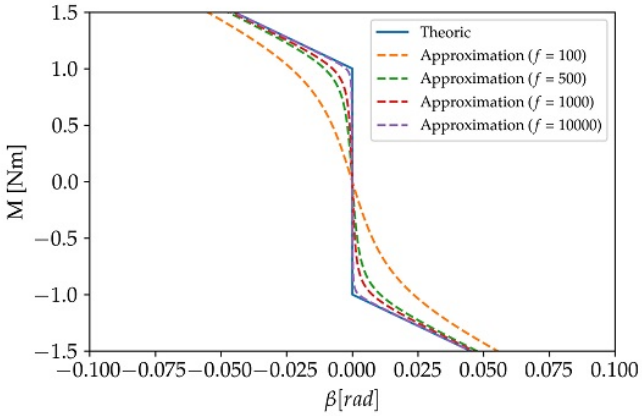


Figure 2.26: Step response of a generic ideal preloaded spring (blue solid line) along with the step response approximated function to varying the approximated factor f [219].

where the r-th generalized active force is defined as follows,

$$F_r = \sum_{j=1}^{N_b} \left(\frac{\partial \boldsymbol{v}_j}{\partial q_r} \cdot \boldsymbol{F}_j + \frac{\partial \boldsymbol{\omega}_j}{\partial q_r} \cdot \boldsymbol{M}_j \right) \tag{2.15}$$

on the other hand, the r-th generalized inertia force is defined as follows,

$$F_r^* = -\sum_{j=1}^{N_b} \left(\frac{\partial \boldsymbol{v}_j}{\partial q_r} \cdot m_j \boldsymbol{a}_j + \frac{\partial \boldsymbol{\omega}_j}{\partial q_r} \cdot (\boldsymbol{I_j} \boldsymbol{\alpha}_j + \boldsymbol{\omega}_j \times \boldsymbol{I}_j \boldsymbol{\omega}_j) \right) \tag{2.16}$$

where $\boldsymbol{v}_j$ and $\boldsymbol{\omega}_j$ are the linear velocity of the j-th body, calculated in its COM, and the angular speed of the j-th body around its rotation axis, respectively. Moreover, $\boldsymbol{F}_j$ is the vector of the active force while $\boldsymbol{M}_j$ is the vector of the active moment, both acting on the j-th body. Additionally, $\boldsymbol{\alpha}_j$ and $\boldsymbol{a}_j$ represent the j-th body's angular acceleration about its axis of rotation and the j-th body's linear acceleration computed in its COM, respectively. Finally, the body mass and inertia tensor, are, respectively, m_j and $\boldsymbol{I}_j$. The inertia tensor is expressed in the body's fixed reference frame.

Eq. (2.14) generates a set of n second-order Ordinary Differential Equations (ODEs), which coincides with the system of EOMs of the generic MBS, which has the following form:

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{2.17}$$

In order to put the EOMs into state-space form, an extra state vector must be added so that $\boldsymbol{u} = \dot{\boldsymbol{q}}$. By substituting the just defined relationship, which represents a set of differential kinematic constraints, in Eq. (2.17) and adding to the system the set of these n differential kinematics equations then a new augmented system of ODEs of the first order can be defined, which therefore has size $2n$. This procedure yields a linear system having the following structure:

$$\boldsymbol{A}\dot{\boldsymbol{x}} = \boldsymbol{B}_{rhs} \tag{2.18}$$

where $\boldsymbol{A}$ is referred to as the augmented mass matrix, $\dot{\boldsymbol{x}} = [\dot{\boldsymbol{q}}, \dot{\boldsymbol{u}}]$ is the state vector's time derivative, and $\boldsymbol{B_{rhs}}$ is referred to as the augmented forcing vector, which is also defined as the system's right-hand side term. The right-hand side component, which contains also the Coriolis term, depends on both generalized coordinates $\boldsymbol{q}$ and its temporal derivatives $\dot{\boldsymbol{q}}$, whereas the mass matrix depends solely on $\boldsymbol{q}$. As a system of $2n$ first-order ODEs, it is amenable to simple time integration using any integration schema. In order to balance precision and processing burden, it has been employed an explicit Runge-Kutta scheme of order 5(4) in this instance.

Soil contact model

For the model describing the contacts between the wheel and the soil, it has been elected to combine a dry friction model with a model of the terrain as a hard and elastic material. The wheel is considered a rigid and not deformable body. As a result, the contact which occurs between the bodies may be seen as a contact point because both the terrain and the generic wheel can be thought of as hard bodies. These presumptions indicate that the wheels' grousers have no impact on the model. Furthermore, depending on the shape of the underlying surface, more than one distinguishable point of contact may be present at any given time. The normal contact force together with the frictional force both exert their effects on the system at the contact site. The normal contact force is modeled by using one of the penalty methods for contact dynamics. More precisely, it has been elected to describe the normal contact force according to the non-linear description of the *Kelvin-Voigt* model, in which the contact between each wheel and the underlying terrain is described by a non-linear spring damper parallel system, and is defined as follows,

$$F_n = -k_g p^{\sigma} - c_g \dot{p} \tag{2.19}$$

where p denotes the depth of the wheel's penetration into the underlying terrain, $\dot{p}$ is the temporal derivative of the penetration p, σ is the coefficient accounting for the spring's non-linear behavior, k_g and c_g are the stiffness and the damping coefficients of the soil.

For what concerns the frictional force, this one is described according to the model presented by Makkar et al. [220]. Within this model, the frictional force depends solely on the relative speed that the contacting bodies experience, and is defined as an always continuous and differentiable function as follows,

$$F_k(\dot{q}) = \gamma_1 \left[\tanh\left(\gamma_2 \dot{q}\right) - \tanh\left(\gamma_3 \dot{q}\right)\right] + \gamma_4 \tanh\left(\gamma_5 \dot{q}\right) + \gamma_6 \dot{q} \tag{2.20}$$

where γ_i are defined as positive constants which have to be tuned accordingly, and that allow for the synthetic modeling of a generic frictional model that incorporates static friction term, dynamic friction term, the Stribeck effect, and finally a dissipation term. In this context, the equation in Eq. (2.20) describes

the friction coefficient instead of the frictional force. The actual friction force is therefore calculated using the just modeled friction coefficient together with the punctual normal force which is in turn computed using the previously introduced model.

By defining more meaningful engineering characteristics, for instance, the static friction coefficient μ_s associated with the friction transition speed v_s, which is the speed at which the friction coefficient turns to be the static's, the dynamic friction coefficient μ_k associated as well to the friction transition speed v_k, which is the speed at which the coefficient turns to be dynamic's, it has been possible, by utilizing a Dual-Annealing algorithm, to synthesize the γ_i parameters, and therefore extract a friction model using Eq. (2.20). Moreover, Kane's method does not directly include the normal contact force F_n or the friction force F_f in the formulation of the EOMs. Within this context, those terms are instead applied explicitly to the system's right side as generalized forces. These are defined as follows,

$$\boldsymbol{Q}_i = \boldsymbol{F_i} \cdot \frac{\partial \boldsymbol{v_p}}{\partial \dot{\boldsymbol{q}}} \tag{2.21}$$

A multi-point contact formulation has been developed because, depending on the nature of the underlying soil, the rover's wheels may present several points of contact. With particular attention to Figure 2.27, the centroid of the contact is determined for each area of contact (defined as the intersection between the wheel and the soil), and subsequently, a generalized force is applied to each centroid point as specified in Eq. (2.21). Moreover, in order to account for the elasticity of the rover's wheels and the structural yield of the rover's legs, which are neither explicitly represented by the model, the variable k_g is chosen to be low on purpose. Finally, Table 2.5 reports the parameters which have been used for the wheel-soil contact model as well as the physical and geometric characteristics describing the semi-analytical dynamics model.

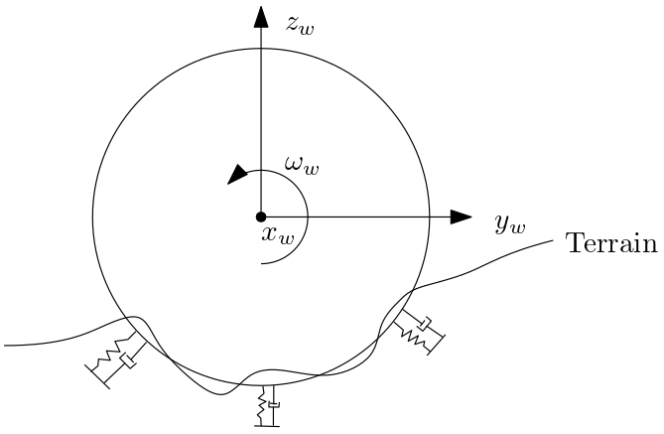


Figure 2.27: Bi-dimensional view of the wheel-terrain interaction modeling as a multi-point contact. On each contact region a spring-damper system is acting between the soil and the wheel [219] .

Table 2.5: Summary of the most representative physical and geometrical parameters chosen, which describe the analytical model of the simplified rover prototype, and the wheel-soil contact interface [219].

Parameter	**Value**	**Parameter**	**Value**
m_R	$1.267\,\mathrm{kg}$	d_{j,w_y}	$0.168\,\mathrm{m}$
m_{SA}	$0.174\,\mathrm{kg}$	d_{j,w_z}	$-0.033\,\mathrm{m}$
$m_{j,1}$	$0.079\,\mathrm{kg}$	$R_{j,w}$	$0.085\,\mathrm{m}$
$m_{j,2}$	$0.393\,\mathrm{kg}$	$M_{0,\tau_{1,1}}$	$1.924 \pm 0.039\mathrm{N\,m}$
$m_{j,w}$	$0.231\,\mathrm{kg}$	$M_{0,\tau_{1,2}}$	$1.336 \pm 0.046\mathrm{N\,m}$
Δ	13.818°	$M_{0,\tau_{2,1}}$	$2.096 \pm 0.089\mathrm{N\,m}$
$L \times W$	$0.720\mathrm{m} \times 0.443\mathrm{m}$	$M_{0,\tau_{2,2}}$	$1.221 \pm 0.031\mathrm{N\,m}$
l_{SA}	$0.111\,\mathrm{m}$	$M_{0,\tau_{3,1}}$	$1.725 \pm 0.069\mathrm{N\,m}$
$l_{j,1}$	$0.100\,\mathrm{m}$	$M_{0,\tau_{3,2}}$	$1.480 \pm 0.053\mathrm{N\,m}$
$l_{j,2}$	$0.169\,\mathrm{m}$	$M_{0,\tau_{4,1}}$	$2.048 \pm 0.075\mathrm{N\,m}$
d_{j,w_x}	$0.089\,\mathrm{m}$	$M_{0,\tau_{4,2}}$	$1.117 \pm 0.032\mathrm{N\,m}$
g	$9.81\,\mathrm{N\,m\,s^{-2}}$	k_q	$9.91\,\mathrm{N\,m\,rad^{-1}}$
f	1×10^4	c_q	$0.14\,\mathrm{N\,m\,s\,rad^{-1}}$
k_g	$1 \times 10^6\mathrm{N\,m^{-1}}$	c_g	$1 \times 10^2\mathrm{N\,s\,m^{-1}}$
μ_s	0.3	μ_k	0.2
v_s	$5.5 \times 10^{-4}\mathrm{m\,s^{-1}}$	v_k	$2 \times 10^{-3}\mathrm{m\,s^{-1}}$
σ	1.8		

Terrain definition

The terrain that the rover will have to be capable of traversing is determined by either establishing its elevation by a point-by-point procedure across a bi-dimensional grid or by using a collection of control points that represent the level surface. In this second approach, the points are subsequently interpolated over a bi-dimensional grid, defined so as to have appropriate size and resolution, in order to obtain the terrain elevation map. The generated terrains are then saved into a handy *.json* file which will be loaded when the simulation starts. The simulation environment subsequently generates a bi-dimensional grid and an interpolating function over it. This function will be continuously used for the wheel-soil interaction algorithm, which detects the contact regions and calculates the contact forces.

2.3.5 Simplified kinematics: Ackermann steering and odometry

In this paragraph are reported the development of the two simplified kinematic models of the four wheels steering rover Archimede, which are needed and implemented for defining the basis of the odometry sub-system, for the control of the real prototype, and for the control of the simulated one as well. The models developed within this section, which substantially differ from the one described in the previous section, are based on the following assumptions: (i) the motion

of the mobile robot is constrained to be planar, i.e. on a horizontal plane; (ii) the length of the legs, i.e. the length from the chassis reference frame to the steering axis, is fixed, which means that the joints of the legs are rigid enough to consider them locked. In reality, and in the real prototype, this will not happen. Indeed, as it has been shown in the previous paragraph, the Archimede rover is composed of four articulated legs, therefore the position of the wheel hub with respect to the rover's chassis may change with time and during the motion of the mobile robot in general. However, in the real prototype, it's not possible to find and extract the relative orientation of the single modules with respect to the others, because that would make necessary additional sensors to be placed in the legs, thus in reality it's not possible to derive the exact kinematics of the leg for each time instant. For this reason, it is assumed that for controlling, driving, and positioning estimation purposes of the real prototype the length of the legs is considered fixed.

In literature can be found different methods that can be used to derive the kinematics of wheeled mobile robots. However, in this case, only two approaches will be used, depending on the use they are intended to be implemented, and they are:

- Vector based approach;
- Ackermann Steering.

The first approach is used exclusively for the *direct differential kinematics* derivation, i.e. given the state of the wheels, it is possible to obtain the speeds of the rover's chassis. This approach, as it will be seen in the following paragraph, is used only for the odometry computation for two main reasons: first, it's not easy to get the state of the wheels starting from the chassis speeds, due to non-linearities introduced by the steering wheels, as it will be seen in the following paragraph, hence a heavy computational minimization or optimization process needs to be used in order to solve the problem; second because it does not include the constraints introduced by the steering lower and upper limits, which as shown in the previous section they are subjected to. In order to take care of these constraints it would be needed to augment the dimension of the system in order to include those constraints for each wheel, thus becoming a full optimization problem reflecting on the overall kinematics calculation process performances.

After exposing these assumptions, the schematic representation of the rover with attention to just one j-th leg model using the just introduced approach can be seen in Figure 2.28. In this context the linear velocities of the rover are defined as $\dot{x_R}$, $\dot{y_R}$, while $\dot{z_R} = 0$; the horizontal projected separation of the wheels from the chassis is instead defined by the vector $\boldsymbol{L_{p,j}}$.

The second approach instead is used for the *inverse differential kinematics*, i.e. given the rover's linear and angular speeds it is possible to compute the corresponding wheels' states. This approach, in contrast to the previously introduced one, is used only for controlling the Archimede rover from a "higher level" perspective.

This model can be seen as a further simplification of the model described in the previous section in which the following constraints are introduced: $z = cost$,

$\dot{z}, \chi, \dot{\chi}, \psi, \dot{\psi}, \phi, \dot{\phi}, \beta_{j,i}, \dot{\beta}_{j,i} = 0$. This is equivalent to considering the rover bodies such as the chassis, bogies, and leg modules fixed together composing then a single rigid body.

Vector based approach

Within this formulation, the no-lateral slip and pure rolling assumptions are implicitly included. The basis of this approach consists of relating the velocities of the wheel hub with the ones of the rover's chassis expressed in the rover's reference frame $(\mathbf{G}, \bar{\mathbf{e}}_G)$, using a purely rigid bodies approach. For sake of simplicity is then assumed that the set of chassis, bogies, and legs modules are considered to compose a single rigid body.

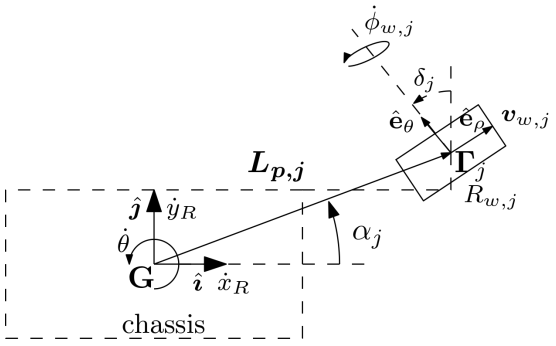


Figure 2.28: Schematic diagram showing the geometric relations between a generic wheel frame and the rover chassis frame.

With reference to Figure 2.28, to the rover body is assigned a new reference frame $(\mathbf{G}, \bar{\mathbf{e}}_R)$, where $\bar{\mathbf{e}}_R = \{\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}, \hat{\boldsymbol{k}}\}$ with $\hat{\boldsymbol{\imath}} = \hat{\mathbf{e}}_{G_x}$ and $\hat{\boldsymbol{\jmath}} = -\hat{\mathbf{e}}_{G_y}$. To the generic wheel of the j-th leg is assigned a new reference frame $(\boldsymbol{\Gamma}_\mathbf{j}, \bar{\mathbf{e}}_W)$ where $\bar{\mathbf{e}}_W = \{\hat{\mathbf{e}}_\rho, \hat{\mathbf{e}}_\theta, \hat{\mathbf{e}}_\phi\}$. The reference frame assigned to the j-th wheel is constructed in such a way that when the wheels are not steered the versor $\hat{\mathbf{e}}_\rho$ points towards the right side of the page, while the versor $\hat{\mathbf{e}}_\theta$ points towards the top side of the page. Moreover, within this redefinition of the frames the j-th wheel rotates around the $\hat{\mathbf{e}}_\theta$ with an angular speed $\dot{\phi}_{w,j}$, hence $\dot{\phi}_{w,j} = -\dot{\zeta}_j$, while the steering angle coincides with δ_j. The linear speeds of the rover, expressed in its reference frame $(\mathbf{G}, \bar{\mathbf{e}}_R)$, are $\dot{x}_R$ and $\dot{y}_R$ while $\dot{z}_R = 0$. The vector $\boldsymbol{L_{p,j}}$ describes the position of the j-th wheel's reference frame with respect to the rover's body reference frame. Moreover, since the legs have been considered to be rigid, together with symmetric considerations, the modulus of this vector is $||\boldsymbol{L_{p,j}}|| = L_p = \frac{L}{2} + \frac{W}{2}$. Hence, still, with reference to Figure 2.28, it is possible to establish a relation between the linear velocity of the j-th generic leg, and the linear velocity of the rover's body as follows,

$$\boldsymbol{v}_{w,j} = \boldsymbol{v}_\mathbf{G} + \dot{\theta}\hat{\boldsymbol{k}} \times (\boldsymbol{\Gamma}_j - \mathbf{G}) \tag{2.22}$$

Referring now to Figure 2.29, and repeating the process just illustrated for the single generic j-th wheel also for the remaining wheels, and by separating

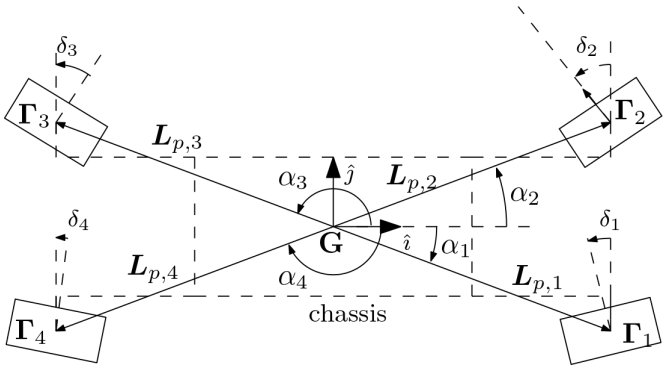


Figure 2.29: Schematic diagram of the rover with its four legs and wheels.

the appropriate components of each of the equations, it is, therefore, possible to obtain a matrix relation having the following form,

$$\boldsymbol{A_k}\dot{\boldsymbol{x}}_R = \boldsymbol{B_k} \tag{2.23}$$

where $\dot{\boldsymbol{x}}_R \in \mathbb{R}^{3\times 1}$ is a vector defining the speed state of the rover's body, that is $\dot{\boldsymbol{x}}_R = [\dot{x}_R, \dot{y}_R, \dot{\theta}]$. The subscript k is used in this context to differentiate the matrices used for the kinematics computation, from the ones describing the dynamics model and obtained in the previous paragraph. The matrix $\boldsymbol{A_k}$ is, instead, expressed as follows,

$$\boldsymbol{A_k} = \begin{bmatrix} 1 & 0 & -L_p\sin(\alpha_1) \\ 1 & 0 & -L_p\sin(\alpha_2) \\ 1 & 0 & -L_p\sin(\alpha_3) \\ 1 & 0 & -L_p\sin(\alpha_4) \\ 0 & 1 & L_p\cos(\alpha_1) \\ 0 & 1 & L_p\cos(\alpha_2) \\ 0 & 1 & L_p\cos(\alpha_3) \\ 0 & 1 & L_p\cos(\alpha_4) \end{bmatrix} \tag{2.24}$$

on the other hand, the matrix $\boldsymbol{B_k}$ is expressed as follows,

$$\boldsymbol{B_k} = \begin{bmatrix} R_{1,w}\dot{\phi}_{w,1}\cos(\delta_1) \\ R_{2,w}\dot{\phi}_{w,2}\cos(\delta_2) \\ R_{3,w}\dot{\phi}_{w,3}\cos(\delta_3) \\ R_{4,w}\dot{\phi}_{w,4}\cos(\delta_4) \\ R_{1,w}\dot{\phi}_{w,1}\sin(\delta_1) \\ R_{2,w}\dot{\phi}_{w,2}\sin(\delta_2) \\ R_{3,w}\dot{\phi}_{w,3}\sin(\delta_3) \\ R_{4,w}\dot{\phi}_{w,4}\sin(\delta_4) \end{bmatrix} \tag{2.25}$$

$\boldsymbol{A_k} \in \mathbb{R}^{8\times 3}$ is a matrix which in this specific case, and under the aforementioned assumptions, is constant in time, and depends only on the geometry of the rover. On the other hand, the matrix $\boldsymbol{B_k} \in \mathbb{R}^{8\times 1}$ contains the wheels' state parameters, i.e. the wheels' rotational speed and the steering angles. It must be pointed out

that if the aforementioned assumptions wouldn't have been made, the $\boldsymbol{A_k}$ matrix would change in time, since the geometry of the rover basically changes during motion, resulting in the fact that the structure of the matrices would be way more complex.

Eq. (2.23) represent a linear system described by eight equations and three variables. This means that the system is over-defined, and five equations must be linearly dependent on the other three. However, in this specific case, the over-defined system will be directly resolved, without identifying and excluding the five linearly dependent equations. The $\boldsymbol{A_k}$ matrix, as seen above, is not squared, thus not directly invertible. Therefore, in order to isolate $\dot{\boldsymbol{x}}_R$, thus inverting $\boldsymbol{A_k}$, the Moore-Penrose left pseudoinverse will be used. Hence,

$$\dot{\boldsymbol{x}}_R = (\boldsymbol{A_k^T A_k})^{-1} \boldsymbol{A_k^T B} \tag{2.26}$$

Eq. (2.26) can be also expressed in the inertial reference frame by knowing the transformation operator $\boldsymbol{R}$ which relates the rover's reference frame with respect to the inertial one. Hence,

$$\dot{\boldsymbol{x}}_I = \boldsymbol{R^T} \dot{\boldsymbol{x}}_R \tag{2.27}$$

where $\boldsymbol{R}$ is the rotation operator in the bi-dimensional case:

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

Eq. (2.26) and Eq. (2.27) are the equations that are at the base of the odometric positioning estimation system which has been coded and included in the firmware of the Archimede rover prototype. The same is implemented in the odometry sub-system implemented as a C++ plugin in the Gazebo simulation environment illustrated in the following paragraphs. Specifically, the odometric system of the Archimede rover is based on reading at high frequencies the encoders of the eight motors, i.e. both the steering and driving ones. More precisely, by discretizing the time t, it is possible to estimate the rover's position in the inertial frame at the time frame t_k as follows:

$$\boldsymbol{x}_I(t_k) = \boldsymbol{x}_{I,0} + \sum_{i=0}^{k} \boldsymbol{R}(t_i)^{\boldsymbol{T}} \dot{\boldsymbol{x}}_R(t_i) \Delta t(t_i) \tag{2.29}$$

where $\boldsymbol{x}_{I,0}$ represents the initial condition of the rover in the inertial frame, and it is assumed to be non-null in general.

Ackermann steering

The purpose of this section is to provide an overview of an approach to vehicle steering, known as Ackermann steering. This approach makes it possible to derive the expression of the inverse differential kinematics for a vehicle in general in a very simple manner. The approach is based on purely geometric concepts and is derived on the basis of two fundamental assumptions: the first is that the vehicle

follows a purely rigid motion, and the second is that the two constraints imposed by the wheels are always respected at every instant of time. The constraints are that of pure rolling and that of no lateral slip. Considering a generic rotating vehicle's wheel, and taking into account the above assumptions, it follows that the constraint of no lateral slip defines a line of zero transverse velocities and that the wheel follows an instantaneous linear velocity parallel to the plane and in the direction of the wheel's advance. In general, the motion of a rigid body can be described by either a motion of pure translation or by a motion of pure rotation around a so-called *Instantaneous Centre of Rotation* (ICR). It follows, that under the above assumptions, for the generic wheel the axis of rotation coincides with the locus of the points of the ICR, which remains undefined for just one wheel. By adding another wheel, therefore, the ICR is completely defined and coincides with the point of intersection of the wheel rotation axes. It follows that once the ICR is defined in the space, all the points of the vehicle must be subject to a purely rotary motion. Therefore, from its knowledge it is possible to establish, instant by instant, the linear velocities of each point and the steering angles so that the condition of Ackermann's steering is respected, i.e. that no lateral slippage occurs on the wheels and that they rotate without slipping. Finally, it must be pointed out that the respect of the two conditions has the effect of not introducing large reaction stresses to the vehicle's structural elements.

This approach is well known in the automotive field [221, 222], due to its simplicity. The approach is actually implemented in most of the common cars [223, 224], the only exceptions are high-performance cars, e.g. Formula 1 cars, where the dynamic components become predominant and a variant of Ackermann steering called Anti-Ackermann is used. This approach is not used only for common cars, but it is rather studied and used in every kind of vehicle. In fact, as instance, it has been applied to 8x8 special vehicles [225], to four-wheel steering vehicles [226,227], or applied to logistic trains, i.e. systems composed by a tractor and a series of trailers [228].

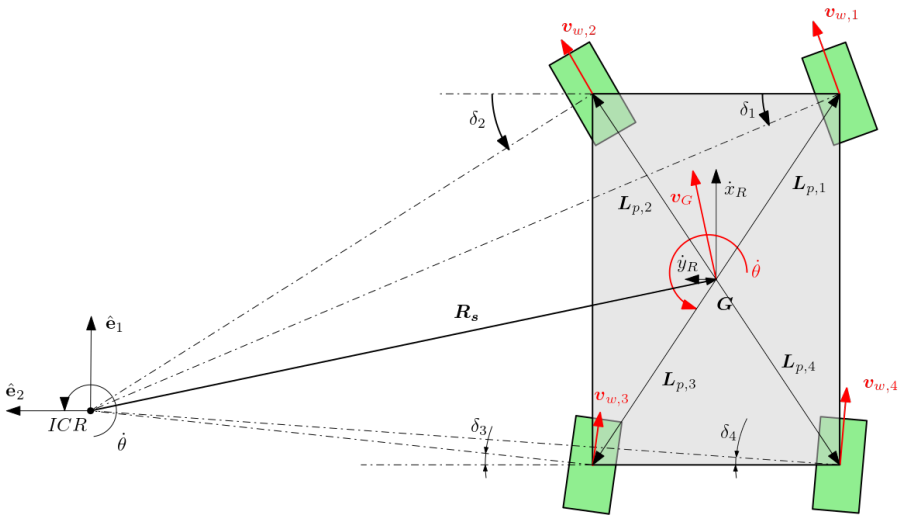


Figure 2.30: General Ackermann steering schematic representation for a four-wheel steering vehicle.

With reference to Figure 2.30, it is apparent that by assigning a tangential velocity to the center of the rover and an angular velocity it's possible to determine the ICR. In fact, by placing a convenient reference frame in the ICR location, it follows that:

$$\boldsymbol{v}_G = \dot{\theta}\hat{\mathbf{k}} \times \boldsymbol{R}_s \tag{2.30}$$

By developing the cross product in Eq. (2.30) and separating the appropriate components it is possible to derive the position of the center of the rover ($R_{s,x}$, $R_{s,y}$) with respect to the ICR, or equivalently the position of the ICR with respect to the rover reference frame. It follows that the position of the ICR is a function that depends exclusively on the linear and angular speeds commanded to the rover.

Once the location of the ICR has been computed, it is possible then to determine the linear velocities of the single generic j-th wheel as follows,

$$\boldsymbol{v}_{w,j} = v_{w,j_x}\hat{\imath} + v_{w,j_y}\hat{\jmath} = \dot{\theta}\hat{\mathbf{k}} \times (\boldsymbol{R}_s + \boldsymbol{L}_{p,j}) \tag{2.31}$$

while the steering angle for the single generic j-th wheel is computed as:

$$\delta_j = \arctan\left(\frac{v_{w,j_y}}{v_{w,j_x}}\right) \tag{2.32}$$

It is apparent that this approach makes it very simple to derive a formulation of the inverse differential kinematics for a general vehicle.

2.3.6 ICR projection approach

As seen the full kinematics of the rover is complex, moreover it includes lower and upper limits for each steer axis, which leads to an increase in the complexity of the kinematics problem. In principle, (2.23), developed for direct differential kinematics, could be used for inverse differential kinematics as well. However, as mentioned above, resolving the system such to extract the wheels' speed and the steering angles is not straightforward. The steering angles are arguments of non-linear functions, and this would require to implement of root-finding algorithms, in order to find a solution that $min||\boldsymbol{A}_k\dot{\boldsymbol{x}}_R - \boldsymbol{B}_k||$. Moreover, to take into account the joint limits' lower and upper constraints it is needed to encode these constraints and augment the dimension of the system that needs to be solved. It follows that implementing such methods is computationally inefficient, especially if they need to be integrated into a control loop, which needs to be executed at a high frequency. Accounting for all these considerations, it had been decided to use the general Ackermann steering formulation, hence Eq. (2.30), Eq. (2.31) and Eq. (2.32), for the inverse differential kinematics, while (2.23) for the direct differential kinematics for the computation of the odometry.

Using the general Ackermann formulation, illustrated in the previous section, raises the need to take into account the lower and upper limits of the steering joints of the rover's wheel hubs. While for the general Ackermann steering, considering a vehicle not subjected to any steering limitation, the ICR can be placed

anywhere in the domain, i.e. ($x_{ICR} \in [-\infty, \infty]$ and $y_{ICR} \in [-\infty, \infty]$); it is apparent that for a vehicle subjected to steering limitations, the ICR must be constrained to belong to specific locations of the plane. From now on the ICR point will be represented by the symbol $\mathbf{\Omega}_r = [x_{ICR}, y_{ICR}]$.

In particular, by considering a generic j-th wheel, it has been seen that under the Ackermann assumptions $\mathbf{\Omega}_r$ lies along the driving axis of the wheel. If now it is assumed to perform a sweep from $\delta_{j,min}$ to $\delta_{j,max}$ the wheel axis draws an unlimited triangular surface S_j with the wheel's steering axis as its vertex. This region identifies all possible permissible configurations for $\mathbf{\Omega}_r$ for j-th wheel. In addition, considering the opposite direction of the wheel's axis of advance, another region is generated that is symmetrical to the first and mirrored. If we now introduce another wheel i to the system and repeat the procedure done for wheel j, it generates its own triangular surface S_i, which is valid for $\mathbf{\Omega}_r$ and for the wheel i. Remembering that $\mathbf{\Omega}_r$ must be absolutely in common with both wheels, it follows that the ICR eligibility region must be a surface in common with both wheels. This means, that the eligibility region of $\mathbf{\Omega}_r$ is derived by intersecting the individual surfaces S_i and S_j, hence $S_i \cap S_j$. These considerations and the procedure just indicated, can be seen in graphical form in Figure 2.32.

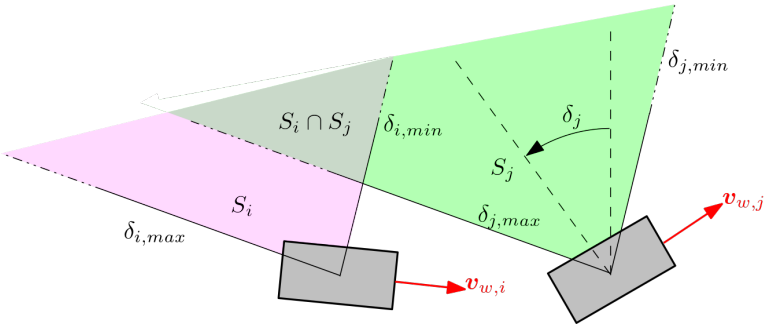


Figure 2.31: Graphical representation of the admissible ICR location for the case of two wheels.

Table 2.6: Summary of the Archimede wheel hub's steering joints lower and upper limitations.

Wheel ID	δ_{min}	δ_{max}
1	33.7°	−93°
2	93°	−33.7°
3	33.7°	−93°
4	93°	−33.7°

Consider now the Archimede rover, which is a vehicle that has four steered wheels each subject to lower and upper limits on the steering joints, the values of which are given in Table 2.6. Performing the process that was described earlier for generic wheel j, and iterating it for each wheel of the rover yields a set of surfaces where $\mathbf{\Omega}_r$ is allowable for each wheel, and thus for the entire system.

It can be demonstrated that the surfaces that have been obtained are five in number, and they are all disconnected from each other. Specifically, a triangular surface was identified that runs along the positive semi-axis of the rover's y axis, an equal one that runs along the negative semi-axis, a surface with four sides located below the belly of the rover, and two very narrow surfaces that run along the positive and negative semi-axes of the rover's x axis. However, the latter two areas are placed very far from the center of the rover and are those permissible areas of $\mathbf{\Omega}_r$ that allow lateral guidance of the rover. The admissible zones for $\mathbf{\Omega}_r$, which were obtained for the Archimedes rover, can be seen in graphical form in Figure 2.32. However, the figure shows only the central surface and the two large surfaces located on the sides of the rover. The two front and back surfaces are not depicted because they are placed far apart.

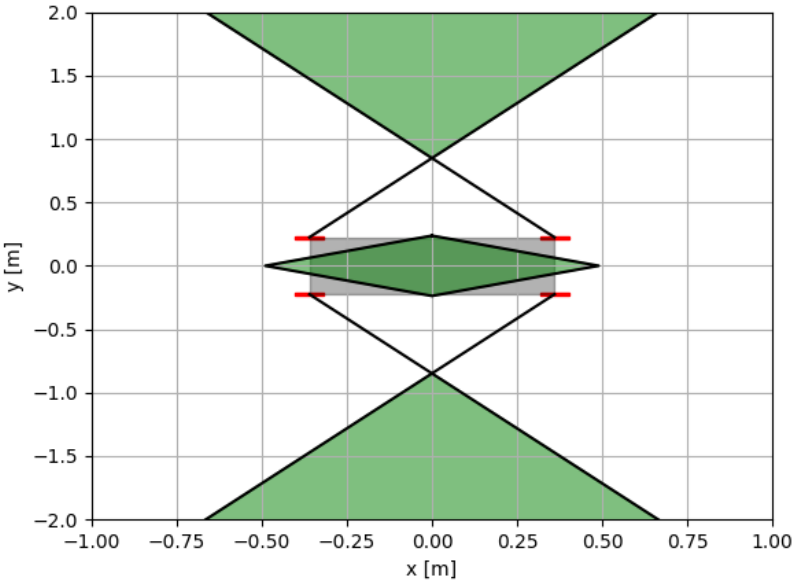


Figure 2.32: View of the admissible ICR surfaces obtained for the rover Archimede: the laterals and the central ones.

We will denote from now on the set of admissible surfaces of $\mathbf{\Omega}_r$ as S^+, and the set of ineligible surfaces as $S^- = \mathbb{R}^2 \setminus S^+$. From the point of view of navigation algorithms, path planning, and high-level control in general, it is very convenient to command the linear and angular velocities of the rover, and thus through inverse differential kinematics map these inputs to the velocities and steering angles of the individual wheels. The differential kinematics that has been decided to use, i.e., Ackermann steering, from the linear and angular velocity inputs, goes through $\mathbf{\Omega}_r$ calculation, through which it then performs the mapping of the rover's linear and angular velocities. However, it has just been seen that not all of the two-dimensional domain is allowable for $\mathbf{\Omega}_r$ for the rover under consideration, and Ackermann kinematics does not introduce any method that ensures that $\mathbf{\Omega}_r \in S^+$ for each instant of time. So, it may happen that for some commanded control signal, the calculated $\mathbf{\Omega}_r$ is invalid and the constraints are therefore violated.

It is, therefore, necessary to develop an approach that makes sure that regardless of the velocity inputs commanded to the rover, $\boldsymbol{\Omega}_r$ always falls within S^+. For this purpose, a method is introduced that is based on projecting the ICR into the admissible zones. Specifically given a general input command, a $\boldsymbol{\Omega}_r$ point is obtained, a new ICR point $\boldsymbol{\Omega}_r^*$ defined as follows is then calculated:

$$\boldsymbol{\Omega}_r^* = \begin{cases} \boldsymbol{\Omega}_r, & \text{if} \quad \boldsymbol{\Omega}_r \in S^+ \\ \boldsymbol{P}_{pr}, & \text{if} \quad \boldsymbol{\Omega}_r \in S^- \wedge \boldsymbol{P}_{pr} \in S^+ \wedge ||\boldsymbol{P} - \boldsymbol{\Omega}_r||_{min} \end{cases}$$

By using this new approach based on the Ackermann steering and on the projection of the ICR, it is always ensured that the ICR, regardless of the control signals arrived as input to the kinematic block, always falls within the permissible zones. Therefore none of the constraints imposed on the steering axes are ever violated.

One very important thing to mention is that through the use of Ackermann steering and the ICR projection method, it is also possible to identify and derive different driving modes of the Archimedes rover. The most characteristic and interesting rover driving modes that have been identified are eight and can be listed as follows,

Mode A **Car-Like**: This driving mode can be observed in Figure 2.33a. In this driving mode, the steering joints of the rear wheels are locked, hence the rover ICR is constrained to pose along the driving axis of the rear wheels of the rover. In this configuration then only the front wheels are allowed to steer while the rear ones are locked, making the Archimede rover similar to a common car. It is apparent that the locus of the admissible ICR are not surfaces anymore, but are a subset of S^+, i.e. lines coincident with the rear driving wheel's axes;

Mode B **Symmetric Ackermann Steering**: This driving mode can be observed in Figure 2.33b. This driving mode is very similar to the previous one and it differs from it only because in this specific driving mode the ICR of the rover is constrained to be along a line passing through the origin of the rover and coincident with its y-axis. It is worth pointing out that in this configuration the inner wheels to the corner turn more (both with the same angle but in opposite directions) and rotate with slower driving speed; while the outer wheels turn less (both with the same angle but in opposite direction) and rotate with more driving speed.;

Mode C **In place rotation**: This driving mode can be observed in Figure 2.33 c. This driving mode represents a degenerate case of the general Ackermann steering, i.e. the ICR is unique and coincides with the center of the rover. In this driving mode, the rover is characterized by not having any linear speed, while the wheels are arranged in such a configuration that allows the rover to rotate on the spot. It is worth mentioning that in this specific case, the wheels all rotate with the same driving speed.

Mode D **Lateral Drive**: This driving mode can be observed in Figure 2.33 d. In this driving mode, the ICR formally doesn't exist. In practice, it's located somewhere along the x axis of the rover frame and very far away from it. It follows that the wheels are turned by 90° around their steering axis, thus allowing the rover to move sideways. In this configuration, the rover doesn't have any angular velocity or any linear speed along the x axis, and the wheels rotate with the same driving speed.

Mode E **Parallel Drive**: This driving mode can be observed in Figure 2.33e. This case is very similar to the previous one because the ICR doesn't exist. The wheels are characterized by the fact that they turn with the same steering angle γ, thus allowing the robot to move in a parallel way. It follows that the rover has only the x and y components of the linear speed and while it doesn't have any angular speed, hence it doesn't turn. Moreover, the wheels rotate at the same driving speed.

Mode F **Outer Ackermann**: With reference to Figure 2.32, this driving mode has been derived by considering in the set of the admissible ICR surfaces S^+ only the two large placed on the side of the rover. It is apparent that the two cases of the Car-Like and Symmetric Ackermann are particular cases of this bigger driving mode case. The ICR if it's not valid is then projected into the boundary of this surface, as can be seen in Figure 2.34.

Mode G **Inner Ackermann**: This drive mode is visible in Figure 2.33 f). This drive mode has been derived by considering in the set of ICR S^+ eligibility surfaces only the central surface located under the belly of the rover and which is visible in Figure 2.32. The ICR if it's not valid is then projected into the boundary of this surface. It is also evident that the case of rotation in place is a special case of this drive mode, which is larger.;

Mode H **General Ackermann**: This driving mode has been obtained by combining the previous modes, i.e. by considering every surface belonging to S^+.

Next, a PyPi-style python package has been developed, which can be installed as a library on any machine, and which implements the inverse differential kinematics through the general Ackermann steering[2]. This library also contains within it the development of the ICR projection method to always calculate $\mathbf{\Omega}_r$ that is permissible by the imposed constraints. Also implemented in the package is the functionality that allows changing the drive modes among all those that have been defined above. So, given the speed input control signals it maps and returns the speeds and steering angles of each wheel. Furthermore, a Desktop application, which also contains a graphical user interface, was also developed within the python package to calculate $\mathbf{\Omega}_r$ through the projection method and allows the procedure to be visualized. This application can be seen in Figure 2.34.

[2]The package and the data are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/rover4ws-kinematics.git`

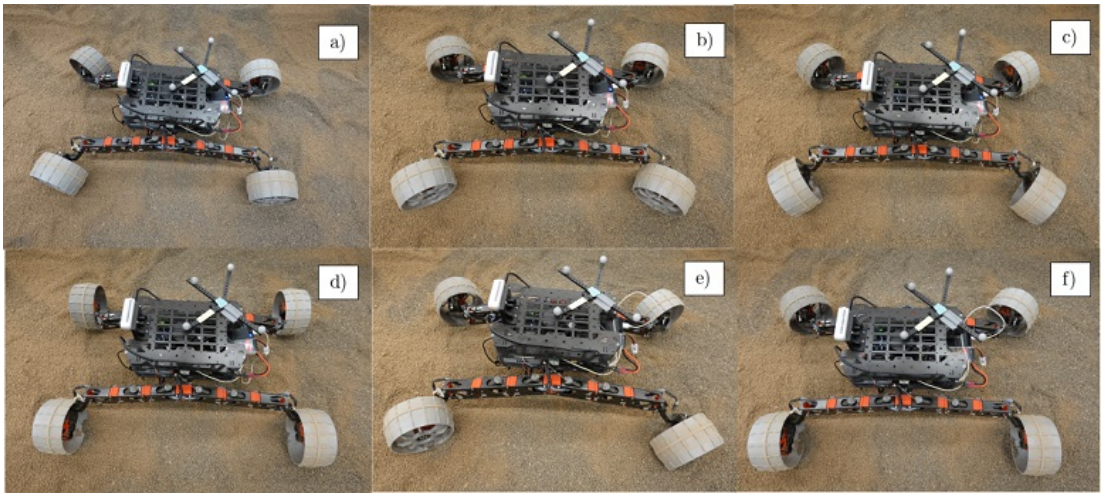


Figure 2.33: Collage of the driving modes identified for the Archimede rover: in a) the car-like, in b) the symmetric steering, in c) the in-place rotation, in d) the lateral drive, in e) the parallel drive, in f) an example of inner Ackermann.

Finally, it has been developed a ROS package for the kinematics of the Archimede rover[3]. This package leverages the Ackermann steering concept and the ICR projection method for joints limits avoidance and it's based on and implements the python library that has been introduced above. The ROS kinematics node basically listens to high-level commands, process them, map them to wheels speed and steering angles, and publishes these quantities in a message on an appropriate topic. Moreover, through a ROS service, it is possible to switch between the different driving modes previously outlined. Finally, the package implements visualization features in RViz of the rover and the admissible surfaces for the ICR location. An example of these visualization features can be seen in Figure 2.35, which shows the rover model together with the outer admissible surfaces (cyan regions), the driving axes of the wheels, and the ICR (green point) projected on the boundary of the left surface.

In addition, several other ROS packages had been developed along with this package and have the aim to allow for teleoperations of the rover from remote input devices. Examples of this are the ROS package for the Logitech joystick[4]and the ROS package for the keyboard[5]. These teleoperation nodes also include the mapping of the driving modes, and the features to switch from one driving mode to another. The nodes basically subscribe to topics over which the data from the input devices come, process them, convert them to high-level control signals of the velocities of the rover, and publish these data to a topic to which the kinematics node subscribes.

[3]The package and the data are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/ros-rover4ws-kinematics.git`

[4]The package and the data are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/robot4ws-teleop-joystick.git`

[5]The package and the data are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/rover4ws_teleop_keyboard.git`

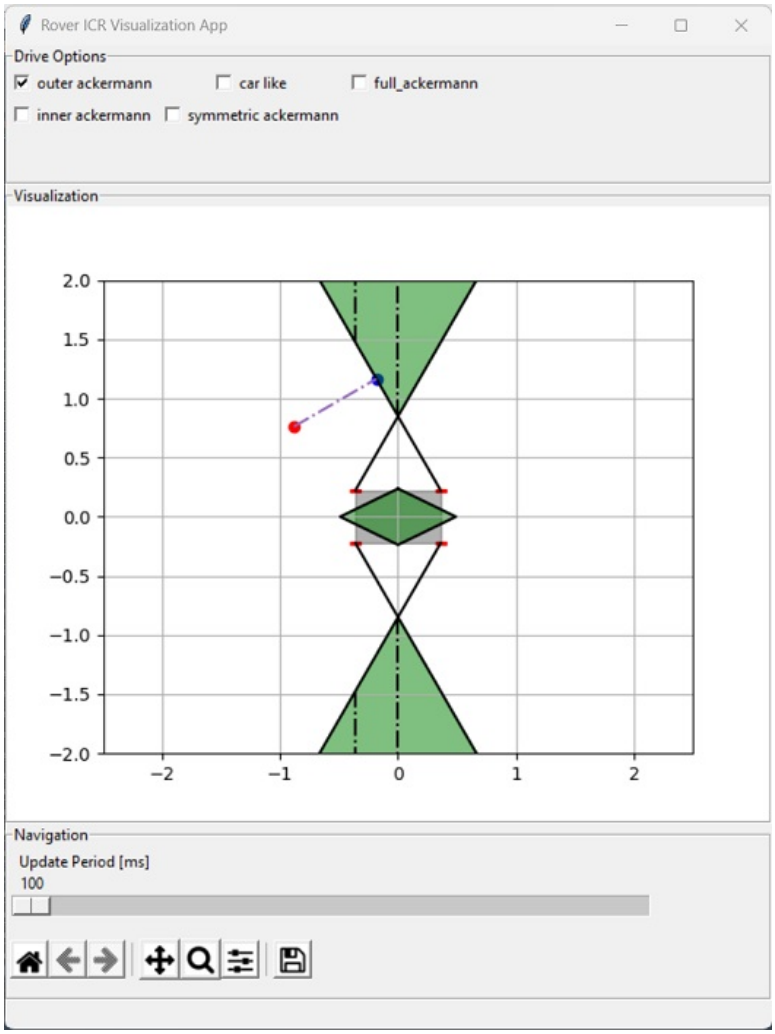


Figure 2.34: ICR projection app, for commanded velocities and ICR projection in the valid areas

2.3.7 ADAMS modeling

In this paragraph, the modeling process of the Archimede rover in the commercial software package ADAMS is presented. This has been done in order to be able to conduct in the next paragraphs a comparison of the results obtained with the simplified analytical model, and therefore validate it, as well as validate both models with the experimental prototype. Within the ADAMS simulation environment, it is modeled the full rover structure, i.e. including the complex S-Structure joints as well as all the internal contacts. The complete and final simulation environment modeled in ADAMS, comprising the Archimede rover and the modeled terrain, can be observed in Figure 2.36a and Figure 2.36b.

More attention is now posed to the details of the modeling process of the rover's MBS. In particular, the initial step in building the MBS representation of the rover Archimede is importing each CAD component that has been created in the commercial software package SolidWorks (see Figure 2.18). Every imported component is then suitably linked to every other component through the proper

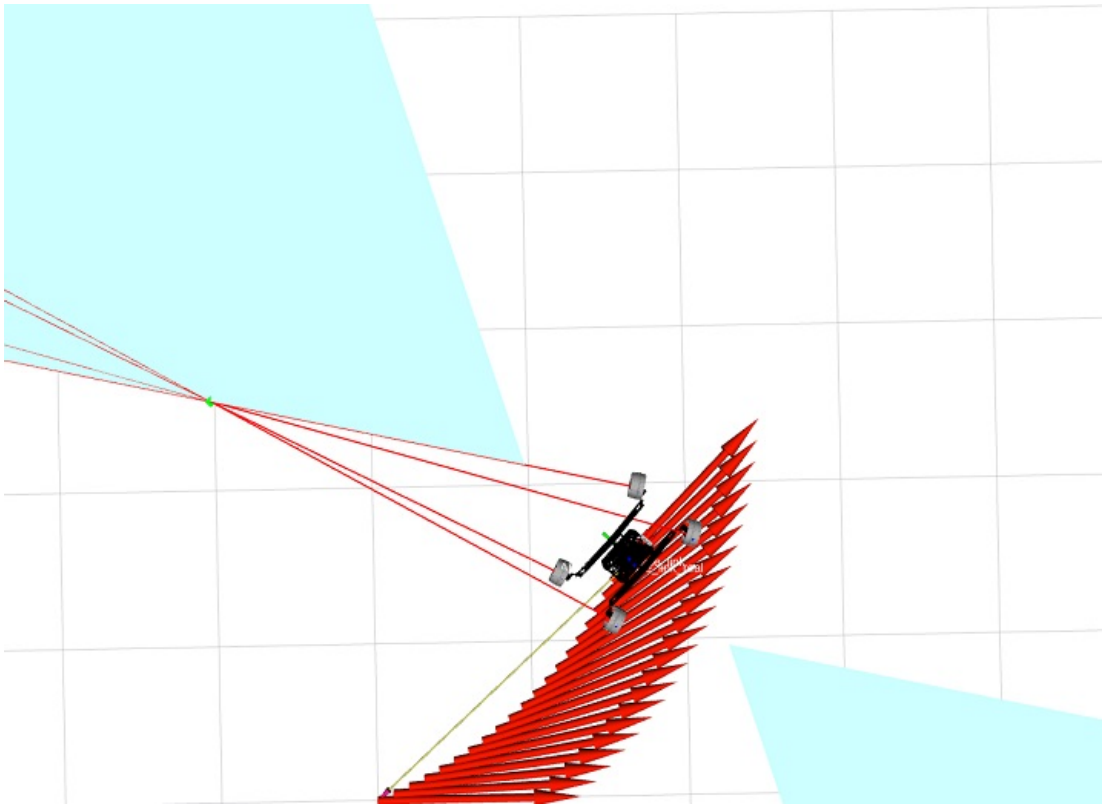
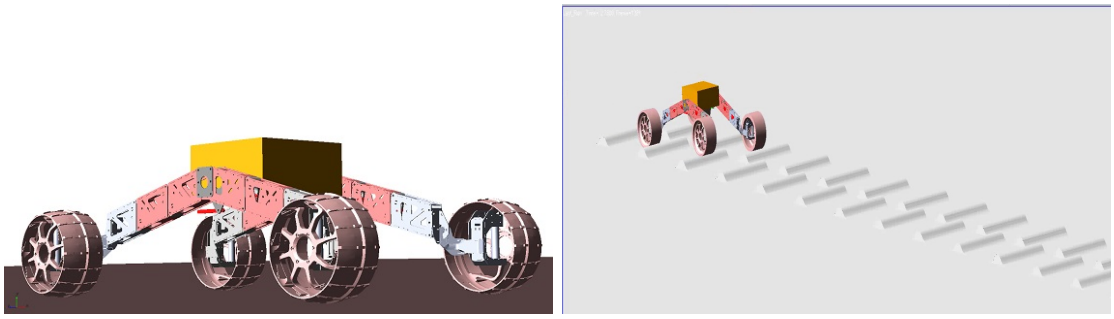


Figure 2.35: RViz visualization feature of the developed kinematics ROS package showing: the model of the rover, the outer admissible surfaces for the ICR, the driving axes of the wheels, and the projected ICR point on the boundary of the left outer admissible surface.



(a) (b)

Figure 2.36: In (a) the CAD model of the Archimede rover was imported in the Adams simulation environment and simplified by relieving it from the non-functional parts [219]. In (b) the rover traverses a road with a series of aligned bumps.

joints. As it has been done in Sec. 2.2, only the functional components are retained, while the rest are considered by appropriately including their mass and inertia properties inside the functional elements. This has been done in order to decrease and lighten the computational load.

Conversely, the basic description of the soil has been modeled as a large, rigid box with the dimensions $(20 \times 20 \times 0.5)\text{m}^3$, to which certain modeled obstacles have been combined at the top, for example, see Figure 2.36b and Figure 2.51b. External loads include the force due to the gravity acceleration that affects the rover MBS, a linear preloaded spring within each S-Structure module of the legs, two contact forces that prevent adjacent modules to penetrate one into the other, and a set of contact forces between every wheel and the ground. Overall, the final model of the Archimede rover is composed of 32 bodies, 32 joints, and 29 forces in the Adams simulation environment.

It is worth pointing out that within the ADAMS simulation environment, it is easy and convenient to model the terrain over which the rover has to operate, even the most complex ones. An example of this can be seen in Figure 2.36b. Indeed, it is just needed to import the CAD model or the STL files into the environment and define contacts between the just imported model and the rover's wheels.

2.3.8 Gazebo modeling

This paragraph aims at presenting the modeling process of the Archimede planetary rover prototype that has been conducted in the Gazebo dynamic simulator. The main idea behind the modeling process is that the overall system should be as close as possible to the real experimental prototype. For this reason, Gazebo and ROS have been chosen, because together they offer a large number of tools and libraries to build a complete robotic system.

Specifically, it has been developed a series of ROS packages that together have the role to take care of setting up and conducting the dynamic simulations of the Archimede rover in Gazebo. The main two packages are the one which describes the model of the rover[6], and the one specific to running the simulations[7].

As mentioned above the first package is responsible for the modeling of the Archimede rover. This has been done by leveraging the ROS package *xacro*, which allows for parametric modeling and the definitions of the so-called "macros". These features are very useful for fast and accurate modeling. The modeling process has been done under one main assumption: in Gazebo what is wanted to achieve is not the accurate description of the dynamics of the rover as could be the case of ADAMS, but rather the accurate description and testing of the single systems composing the rover. For this reason, in the model description ROS package, it has been decided to lock the articulations of the legs of the rover, as well as the relative rotation of the rocker bogies with respect to the chassis. It follows that the resulting model obtained in Gazebo is substantially different from the one modeled in ADAMS, and it's more simple. In fact, it possesses only 14 DOFs, which are: the pose in the space of the chassis, which accounts for 6 DOFs; 4 DOFs for the steering joints, and 4 DOFs for the driving joints. Moreover, since the model of the Archimede rover in Gazebo doesn't possess flexibility of the

[6]The data and the source code are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/robot4ws_description.git`

[7]The data and the source code are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/robot4ws_simulations.git`

legs, thus no suspension system, the motion of the rover is constrained along the horizontal plane. The Gazebo simulation environment instead is modeled either as an empty world, i.e. as a flat surface with no obstacles, or as a large room that can contain also obstacles. Finally, this package contains the definition of which plugin to load within the Gazebo simulation environment. Since the scope of this simulation environment is not focused on the mechanics and dynamics of the rover structure but rather on the simulation and testing of all the sub-systems composing the rover, it has been decided to mimic and simulate every system present on the experimental prototype. It follows that the model of the rover contains plugins that simulate the IMU sensor, the 2D lidar range sensor, the wheels encoders, the Raspberry Pi camera, the RealSense D435i stereo-camera, the kinematics, the actuation of the joints, and finally the odometry subsystem. Moreover, plugins for logging and storing data are implemented as well. A graphical representation of the final rover model in the Gazebo simulation environment is shown in Figure 2.37; while a graphical representation of the description of the connections and relations between the frames, i.e. the single bodies, is shown in Figure 2.39.

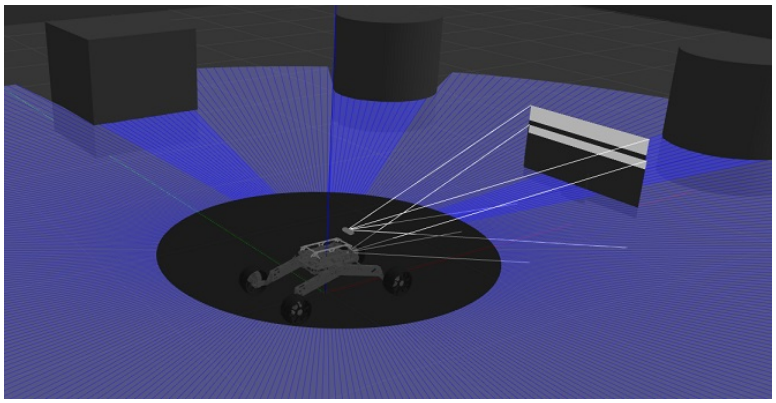


Figure 2.37: Archimede rover in the Gazebo simulation environment.

The second ROS package instead implements the C++ plugins for Gazebo of the rover's kinematics and the odometric sub-system outlined in Sec. 2.3.5, which are also responsible for enabling the ROS-Gazebo co-simulation through the Gazebo ROS bridge, thus making it possible to control the rover from the ROS network. Commands are sent over the network by the user or the control algorithm. The kinematics plugins, which implement separate PID controllers on each controllable joint, receive the sent commands, parse them, and then apply them to the PID joint controllers. On the other hand, the plugin for the odometry sub-system read with a tunable fixed sampling period the "encoders" of the joints, compute the odometry and send the corresponding data over the ROS network. Moreover, this package contains different *.launch* files and each of them is responsible for starting a dedicated Gazebo simulation as well as specific ROS nodes which implement different features needed for the specific simulation. For instance, the most basic launch file will start only the Gazebo simulator with the rover in it and with the Gazebo-ROS bridge enabled; while the most complete launch file will start the Gazebo simulator, the rover model, the ROS kinemat-

ics package implementing the ICR projection, the teleoperations node through remote input devices and the RViz visualization tool. Moreover, having allowed the ROS-Gazebo communication, by connecting MATLAB to the ROS network it had been possible to implement the MATLAB-ROS-Gazebo co-simulation, thus made possible to control the simulated Archimede rover through the same MATLAB application that is used for the experimental rover prototype, since the model of the rover in Gazebo perfectly mimics the experimental one. The overall ROS application used, which implements all the systems actually employed on the experimental prototype, and implements also the MATLAB-ROS-Gazebo co-simulation can be seen in Figure 2.38. This figure illustrates the ROS application in form of a graph, showing the connections between each ROS node, i.e. specific separate processes. It must be pointed out that since the simulated model perfectly mimics the experimental one, it follows that the experimental ROS application graph doesn't differ from the one shown for the case of the simulation setup.

Figure 2.38: Rosgraph of the ROS application for the MATLAB-ROS-Gazebo co-simulation of the Archimede rover.

Figure 2.39: Transformation tree of the Archimede rover in the Gazebo simulation environment.

2.3.9 Odometry subsystem experimental demonstration

This paragraph has the main objective to demonstrate the functioning and capabilities of the odometry subsystem in an experimental setting. All the tests that have been carried out were also performed with the experimental prototype of the rover Archimede, on a non-consistent terrain. The experimental prototype of the Archimedes rover was tested in the sandbox of the PEL laboratory of DLR's Robotik und Mechatronik Institute, located in Muenchen (Germany). The sandbox contains very fine sand that had been chosen as a simulant of lunar dust. For this reason, the design of the wheels of the rover was changed by making them have deeper grousers, and a different shape, in order to provide greater traction capability on this type of sand, and less lateral slippage. Moreover, for every experimental test that had been conducted in the PEL laboratory, the Archimede rover had been tracked by using an ART DTRACK Optical Tracking system, which helps reconstruct the rover's motion. The tracking system software is based on ROS, and data are extracted by leveraging it. A RViz snapshot of the tracking procedure is shown in Figure 2.40, which shows the rover's moving reference frame (Rover's frame) with respect to an inertial frame.

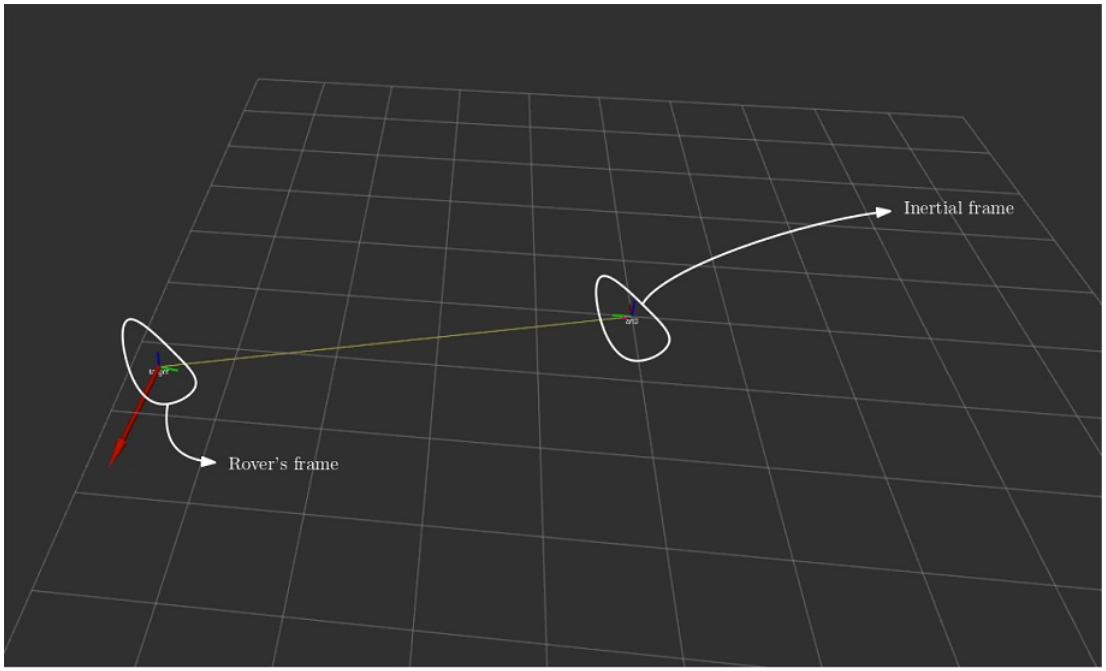


Figure 2.40: RViz representation of the tracking system: rover's frame recreation and relation with the inertial frame.

Additionally, in the PEL laboratory is possible to lift half of the sandbox, thus allowing the generation of all possible soil's surface inclination, in order to evaluate the rover's capability to climb steep slopes. In Figure 2.41 can be clearly seen the experimental setup of the DLR PEL sandbox, the tracking system, the liftable side of the sandbox, and the Archimede rover driving on a 20° slope.

In Figure 2.42a is shown a picture of the Archimede rover while performing fixed beforehand planned path on a leveled surface of loose soil for the odometry

Figure 2.41: Experimental setup of the DLR PEL sandbox and the rover climbing a 20° slope.

sub-system demonstration; in Figure 2.42b is shown a picture of the rover while driving on a slope of 15° of loose soil; in Figure 2.42c is shown a picture of the Archimede rover showing its obstacle negotiation capabilities on a slope of 15° of loose soil over a bed of small rocks.

a) b) c)

Figure 2.42: In (a) the rover performing demonstration of the odometry subsystem on a leveled soil in the sandbox; in (b) the rover climbing a slope; in (c) the rover attempting at negotiating obstacles on a slope of 15°.

Specifically, for the tests performed for odometry subsystem demonstration, it has been elected to evaluate it both experimentally and numerically, in the appropriate Gazebo simulation environment. For the case of the experimental evaluation of the system, driving tests with the Archimede prototype were performed on a horizontal surface, which has been leveled as accurately as possible. It is worth mentioning that given the nature of the terrain, a perfectly horizontal surface is impossible to achieve, thus the surface profile presents changes in the

elevation. Moreover, in the sandbox albeit small there are always potholes and bumps. Driving tests were carried out along four different planned beforehand paths and the motion of the rover was recorded with the optical tracking system. At the end of each test, and for the purpose of comparison, all data on the internal states of the rover were extracted, including the evolution of the odometry and the state of the joints. For what concerns the case of the simulated evaluation of the system, in order to include another method of comparison, it was decided to perform the same tests with the simulated model of the Archimedes rover, in the Gazebo simulation environment, thus using the various ROS packages that were developed and described in the preceding sections.

Specifically, the paths that had been chosen as samples for the odometry subsystem tests demonstration are: (i) a 20° diagonal line in which the rover must track it while driving in parallel mode; (ii) a path drawing a quarter of a circle in which the rover must track it by using the Symmetric-Ackermann driving mode; (iii) a path drawing an "S-like" shape in which the rover must track it by using the Symmetric-Ackermann driving mode; (iv) a path drawing a square shape in which the rover must track it by using a combination of straight and lateral driving modes. For each path type the same tests, with the same driving speed, have been performed both experimentally and numerically. The findings of these tests are reported in Figure 2.43.

From Figure 2.43 we can clearly distinguish the four different trajectories that have been planned. Figure 2.43a shows the outcomes for the 20° diagonal line path; Figure 2.43b shows the same for the quarter of a 1.5 m radius circle; Figure 2.43c shows the same for the s-like path with a radius of 1.5 m; Figure 2.43d shows the same for the case of the squared path. In addition, for each individual figure, i.e., the type of path, the nominal trajectory–nominal trajectory–, the true position of the rover in Gazebo–tracking (gazebo)–, the true position of the experimental rover–tracking (experimental)–, the odometric position of the rover in Gazebo–odometry (gazebo)–, and the odometric position of the experimental rover–odometry (experimental) can be distinguished. From all the figures it can be seen that all the curves preserve the shape of the imposed path and that they are all reasonably close to each other. It can be seen that each curve is subject to errors that can be of various kinds. The most evident ones, which show greater deviation from the nominal paths, are the experimental tracking. Where the biggest deviation is shown for the square path, in which can be seen that the experimental tracking squared shape is highly distorted. This is mainly due to the terrain being far away from being ideal and flat. In fact, the terrain presents bumps and change in elevation which affect the position accuracy. Evidence of this is shown in Figure 2.44, which for each path shows the profile elevation as a function of the path length parameterization. In fact, from the figures, it can be seen that there are differences in elevation which can go from ≈ 5 cm for the case of the quarter of a circle, up to ≈ 30 cm for the case of the square like path.

The sources of errors that cause all those curves to deviate from the nominal ones are many. For example, one source of inaccuracy is given by the nature of the soil, which is loose, which causes the wheels of the rover to be subjected to longitudinal and lateral slippage. Another source of inaccuracy related to the soil

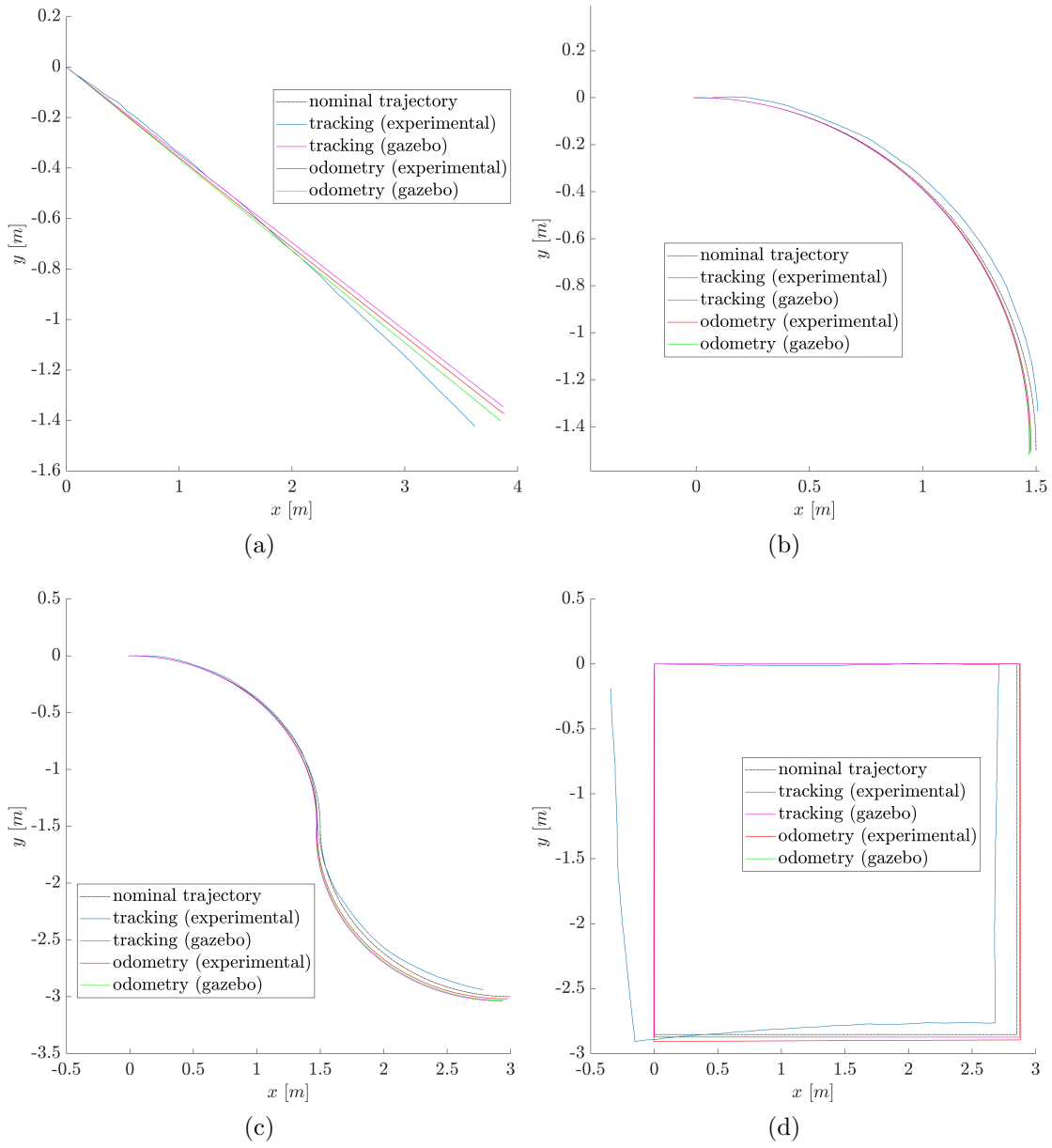


Figure 2.43: Comparison between the real trajectories (tracking) and the position estimation through wheels odometry for (a) the 20° diagonal line; (b) the quarter circle of radius 1.5 m; (c) the s-like path of radius 1.5 m; (d) the square path.

is given by the fact that it is not ideal and not perfectly flat, as already seen. Other sources of inaccuracies can be due to the difference between the effective radius of the wheel and the nominal radius of the wheel, due to grouser presence and sinkage phenomenon. Delays between the sending and the receiving of the commands over the ROS network, which can also be seen from Figure 2.43 and especially from Figure 2.43d, are another source of inaccuracy. Moreover, other sources of inaccuracies are due to numerical errors as such the truncation error caused by the least-square procedure employed to compute the odometric step in Eq. (2.26); or such as the numerical error introduced by the numerical integration

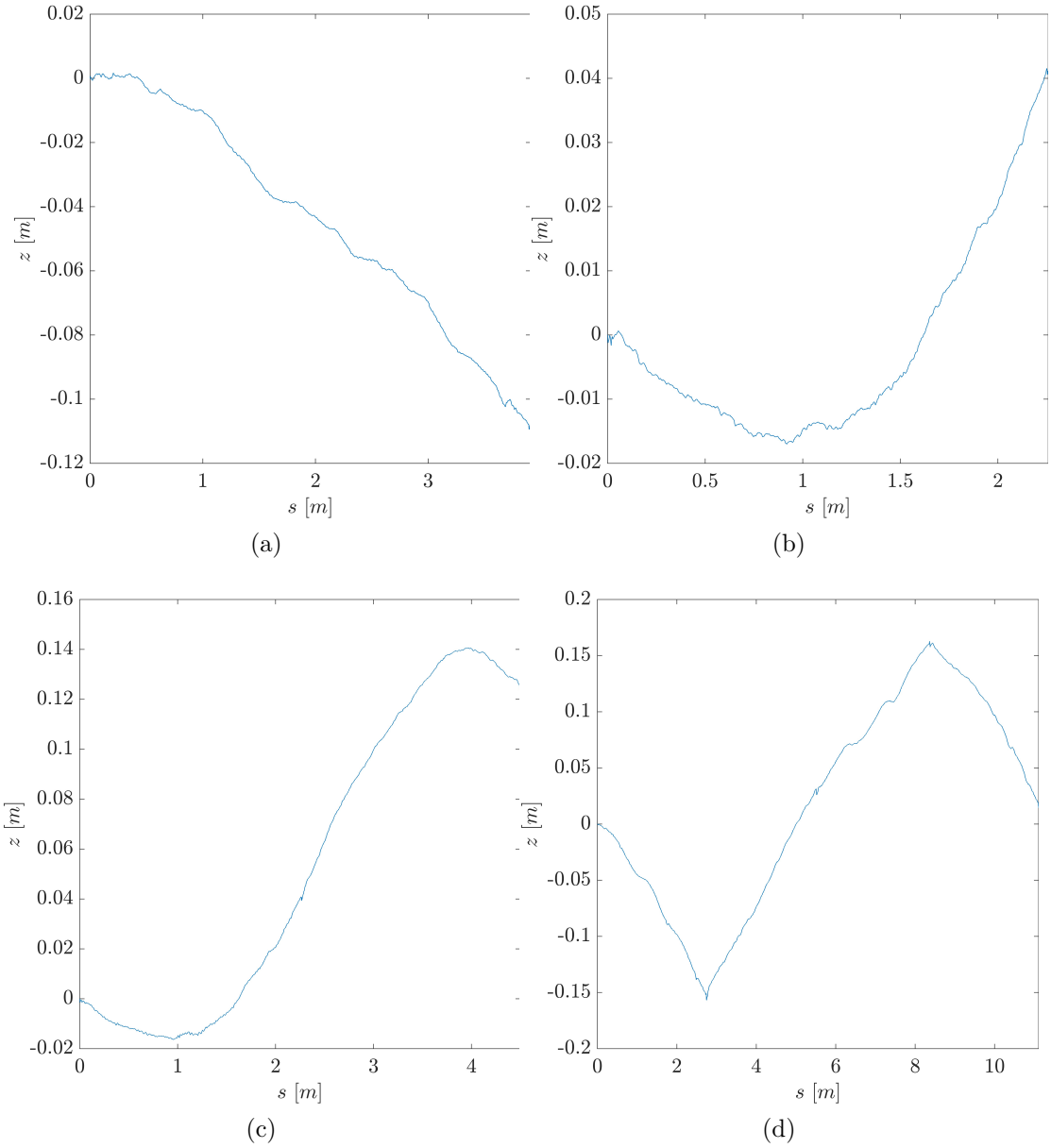


Figure 2.44: Elevation profile in function of the path length parametrization for: (a) the 20° diagonal line; (b) the quarter circle of radius 1.5 m; (c) the s-like path of radius 1.5 m; (d) the square path.

scheme. Other sources of inaccuracies can be due to inertial effects imposed on the system. Finally, since the odometric position estimation is a dead reckoning procedure, it follows that errors tend to accumulate with time, showing greater errors as the traveled distance increases.

In order to establish an evaluation of the accuracy of the odometric position estimation of the rover, it has been elected to compute the end-point trajectory relative position error. This error has been evaluated between the gazebo odometry and gazebo tracking curves, and between the experimental odometry and the experimental tracking curves. The findings are shown as a bar plot in Figure 2.45

for each of the paths performed.

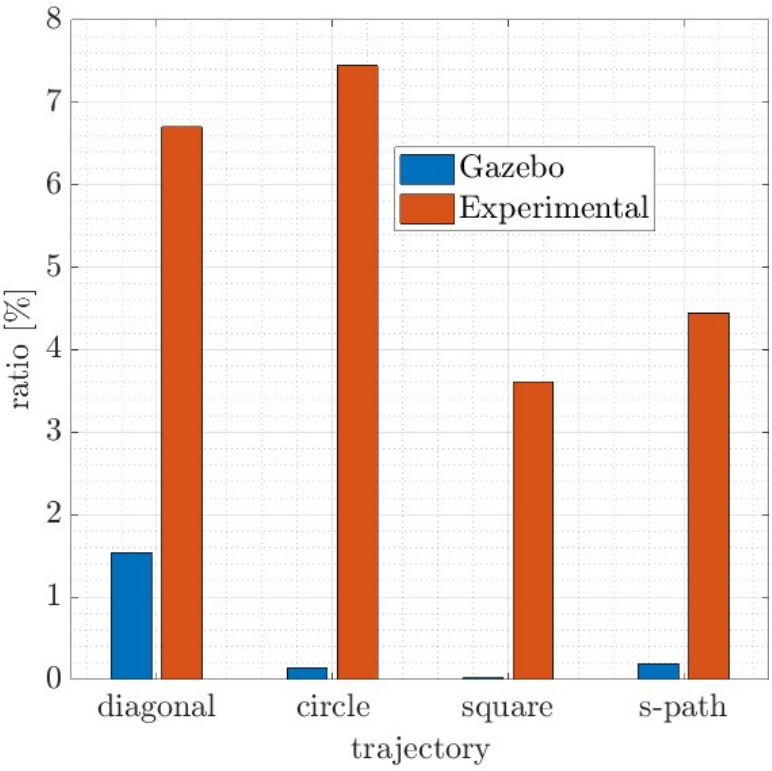


Figure 2.45: End-point relative position error between the odometric position estimation and the real position for all the paths selected and for both numeric and experimental tests.

It can be seen from the figure that for every selected path for both the experimental tests and the numerical tests, the error stays below the 10% threshold. Moreover, it can be seen that the error observed for all the experimental paths are always greater than the ones observed in the numerical simulations. This is because the numerical simulation is a controlled environment, while the experimental case presents many more sources of inaccuracies. The first of these is the inconsistent nature of the terrain and that it is not ideal. In the case of the numerical simulations, the path which exhibits a larger position error is the diagonal path. This can be addressed to the fact that at the beginning of the simulation, all the wheels immediately turn to reach the commanded steer set point. It follows that this fact produces an inertial effect that makes the body of the rover initially spin in the other direction. This has the effect that the rover heading at the beginning of the simulation is not null, thus causing a deviation from the nominal commanded path. This can be also seen from Figure 2.43a.

2.3.10 Dynamics models validation

The goal of this paragraph is to validate the dynamics model which has been obtained by using Kane's method, and which has been presented and described in detail in Sec. 2.3.4. Since the exact EOMs of the proposed analytical model are

numerically integrated using a RK4 integration method, the presented approach is not entirely analytical, this is also because the complex S-Structures joints have been simplified. For this reason and also to differentiate it from the other models the presented analytical model, from now on, will be referred to as the semi-analytical model (SA). Both experimental data obtained by testing on the experimental Archimede prototype and the results of the simulations conducted on the MBS of the rover modeled in ADAMS, as shown in Sec. 2.3.7, are used within the validation process. Finally, this section gives a quick presentation of the elected numerical simulations' setup, while the experimental campaign's setup is outlined after that. Results from experiments and calculations are provided and compared at the end.

Tests definition

Two key case studies are taken into consideration within this paragraph in order to compare the SA model, the complete MBS rover's model in ADAMS, as well as the experimental prototype of the Archimede rover. The selected case studies are:

(Case 1.) *Drop test*: In this instance, the rover is left completely free to fall from a specific high point above the ground. The rover collides with the underlying soil while merely being exposed to gravity acceleration. An attitude nearly horizontal and an attitude significantly skewed are the two starting initial conditions for the rover that are taken into consideration in this scenario. Moreover, within this case has been compared the SA model, the complete ADAMS model, and the experimental prototype.

(Case 2.) *Obstacle negotiation*: The rover is commanded to traverse the horizontal terrain within this second scenario. The rover is positioned such that the right bogie is right behind a modeled obstacle, which is shaped in a way that exerts distinct stresses on each of the rover's DOFs. The obstacle is constructed as a trapezoidal body having sharp edges, measuring 0.1 meters in height, 0.3 meters in width, and 20 degrees in slope. In contrast to the prior case, just the two simulated models are examined and compared, i.e. the SA model and the ADAMS one. Moreover, three distinct rover operating driving speeds have been taken into account in this case: a low speed of $0.34\,\mathrm{m\,s^{-1}}$ and two high speeds of respectively 1 m/s and 2 m/s. Finally, each wheel's angular velocity is actively regulated to maintain the rover's speed, within both the SA and ADAMS models.

Experimental testbed

The preload tension of the linear springs encapsulated in the S-Structures has been measured in order to accurately replicate the experimental testbed inside the simulation environments. The initial detachment tension has been therefore measured using a dedicated load cell. The measured spring preload is then listed

in Table 2.5 as moment preload $M_{0,\tau_{j,i}}$. Figure 2.46 illustrates the drop test experimental setup for the rover prototype. Specifically, cables have been employed in order to secure the rover to an apposite release mechanism that has been 3D printed using the low-friction material POM. The entire system is therefore lifted by using a gantry crane. Moreover, with the aid of a properly calibrated inertial measuring unit (IMU), the attitude of the rover body is regulated by operating the screws on the release mechanism.

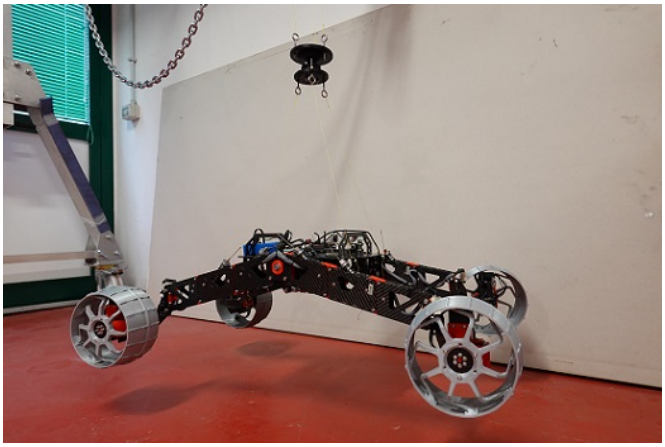


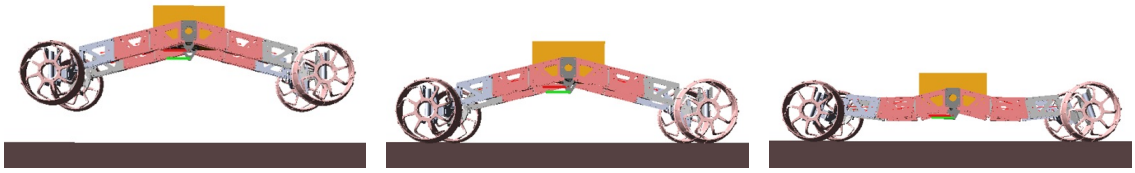
Figure 2.46: Experimental setup for the rover prototype drop test [219].

Results

The outcomes for the three models of the Archimede rover—SA, ADAMS, and experimental—for each of the two test cases previously illustrated—are reported within this section. Reminding that Table 2.5 lists the models' geometrical and physical characteristics. Moreover, it has provided a variety of indexes whose objective are to measure the deviation for each test in order to give a meaningful method for comparing the numerical findings of both approaches with the experimental trial findings. The main characteristics that have been examined are the rover's dynamical behavior during a drop test, in which a rapid collision with the underlying soil occurs, as well as during obstacle negotiation, in which the rover traverses over a small, asymmetric trapezoidal obstacle. Moreover, the introduced indices describe the relative inaccuracy between the experimental prototype and the two considered numerical models.

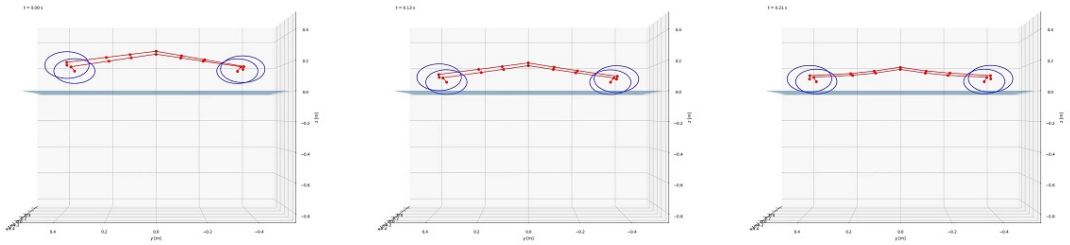
For what concerns the drop test, as has been previously pointed out, the SA, the ADAMS, and the experimental models have been compared. An identical rover initial conditions must be applied to compare all three different models being investigated. For both the horizontal and skewed drop tests, these conditions—which relate to the drop height, the pitch, and the roll angles—are listed in Table 2.7.

The starting position of the rover, the instant it touches the ground, and the moment at which the structure undergoes maximum deflection are each graphically depicted in Figure 2.47a, Figure 2.47b, and Figure 2.47c, respectively, for an example of horizontal drop test simulation in conducted in ADAMS. While



(a) (b) (c)

Figure 2.47: Snapshots of the rover configurations in the different phases of the impact simulation over a horizontal surface in ADAMS: in (a) the initial condition; in (b) the moment of touchdown with the pavement; in (c) the moment of maximum deflection of the overall system [219].



(a) (b) (c)

Figure 2.48: Snapshots of the rover configurations in the different phases of the impact simulation over a horizontal surface with the SA model implemented in python: in (a) the initial condition; in (b) the moment of touchdown with the pavement; in (c) the moment of maximum deflection of the overall system

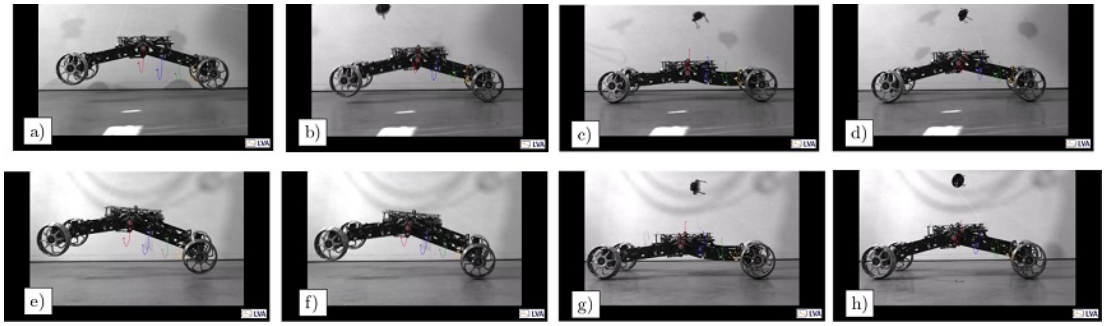


Figure 2.49: Time-series frames of the two experimental drop tests: in the top row the drop test on the horizontal surface; in the bottom row the skewed drop. For both rows the figures in the first column (a and e) represent the initial configuration of the rover; in the second (b and f), the moment of first impact with the ground; in the third (c and g), the moment of maximum displacement of the legs of the rover; in the last (d and h), the final "rest" configuration [219].

the same, for the SA model, are shown in Figure 2.48a, Figure 2.48b, and Figure 2.48c.

In Figure 2.49 are depicted several acquired frames which have been taken by utilizing a high frame rate camera during the rover prototype's experimental drop

Table 2.7: Initial conditions of the rover used in the three models for the horizontal and skewed drop test [219].

Type	z [m]	χ [rad]	ψ [rad]
Horizontal	0.281	0.039	0.0042
Skewed	0.299	0.116	-0.0809

testing. The first row of the figure (a-d) specifically depicts the horizontal drop, whereas its second row (e-h) specifically depicts the skewed one. The frames in each row, from left to right, show the rover's initial configuration, the instant its wheels impact with the surface, the moment of greatest deflection, and its "at rest" posture. In Figure 2.50a and Figure 2.50b, instead, are shown the outcomes in terms of the vertical position of the pin belonging to the rover's left bogie, respectively for the horizontal drop test and the skewed one.

Furthermore, it has been defined an index named deflection error in order to statistically quantify the inaccuracies, which is defined as follows,

$$\varepsilon_{d,*} = \frac{z_{r,*} - z_{min,*}}{z_{r,EXP} - z_{min,EXP}} - 1 \tag{2.33}$$

where $*$ represents the model used, either experimental (EXP), numerical (ADAMS), or semi-analytical (SA), with $z_{min,*} = min(\mathbf{B}_{l,*}|_z)$, and $z_{r,*} = \mathbf{B}_{l,*}|_z(t = 1s)$. The error for the "*at-rest*" configuration, i.e. once the movement of the structure after impact has stopped, can be defined as follows using the same nomenclature:

$$\varepsilon_{r,*} = \frac{\mathbf{B}_{l,*}|_z}{\mathbf{B}_{l,EXP}|_z} - 1 \tag{2.34}$$

In conclusion, it has been defined the at-rest error and the deflection error as two different metrics which will be further used for the quantitative comparison study of all the models. The first is concerned with the sag experienced by the rover when it has stopped moving, while the second is concerned with the displacement of the rover's chassis induced by contact with the ground. The following will describe different features of the model performance using these metrics, furthermore allowing for a thorough examination and comparison.

Differently from the prior case, for the obstacle negotiation case instead just the models of the SA and ADAMS are taken into account and contrasted here. Once more, the initial conditions for both models must be the same in order to compare them. These, in this case, are the initial speed and the positioning of the rover chassis in the simulated environment. The rover's configurations for the case of the velocity equals $1\,\mathrm{m\,s^{-1}}$ are shown in Figure 2.51. More precisely, Figure 2.51a shows the rover's configuration using the SA model while the rover as being intent on traversing the obstacle, with the front-right wheel on top of it. The identical rover configuration is depicted using the ADAMS model in Figure 2.51b, in contrast.

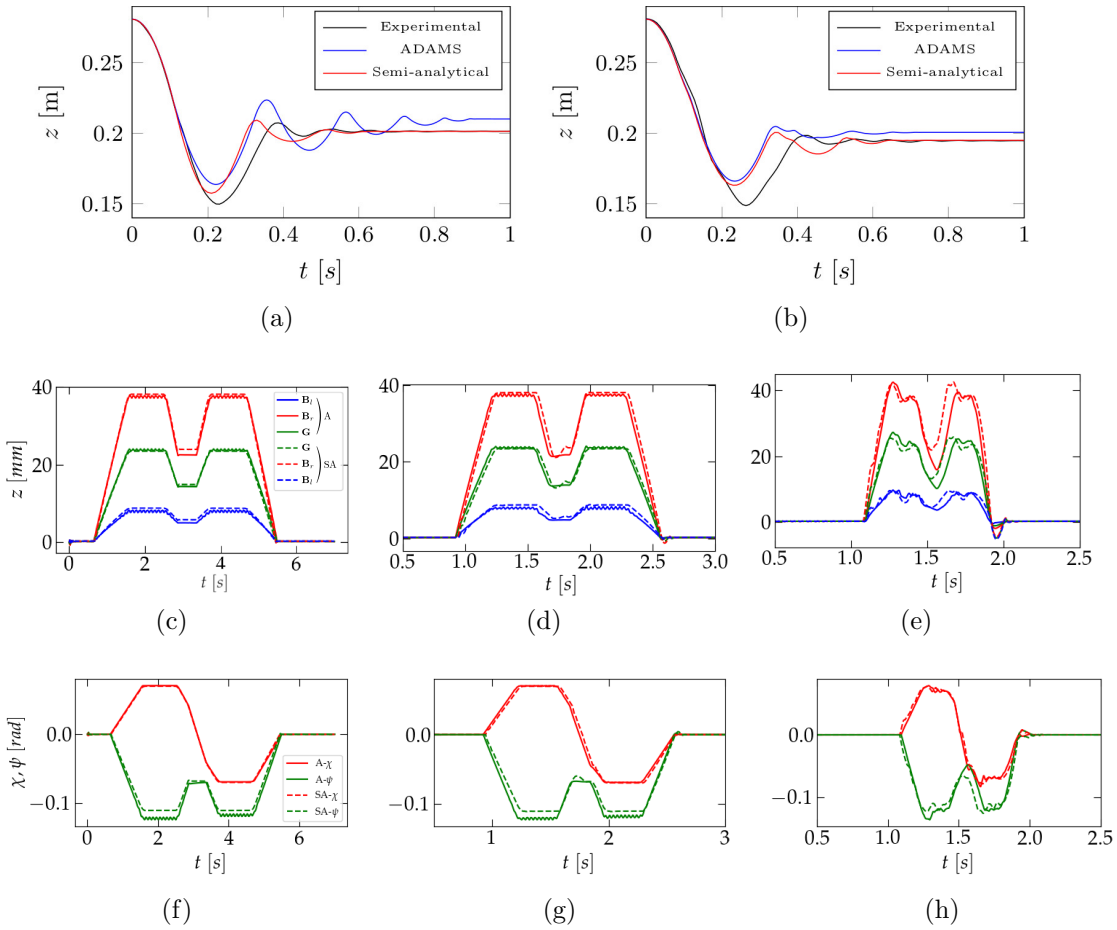


Figure 2.50: In the top row the vertical displacement of the pin of the left bogie of the rover for each of the three models, for the cases of: horizontal drop test (a) and the skewed drop test (b). In the middle row the vertical displacement of the left bogie pin ($\mathbf{B}_l$), the right bogie pin ($\mathbf{B}_r$), and the rover center of mass ($\mathbf{G}$) for the rover longitudinal speed of: 0.3 (c), 1 (d) and 2 (e) $\mathrm{m\,s^{-1}}$. In the bottom row the trend of the rover pitch and roll angles for the SA and Adams models for the rover longitudinal speed of: 0.3 (f), 1 (g) and 2 (h) $\mathrm{m\,s^{-1}}$ [219].

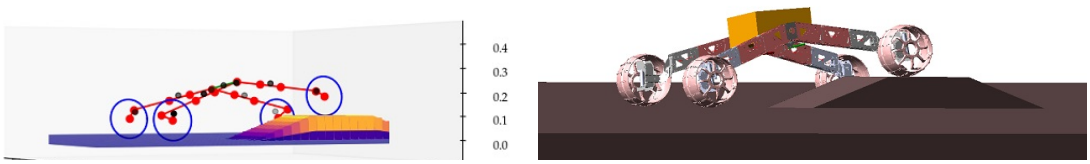


Figure 2.51: In (a) a snapshot of the rover overcoming the ramp obstacle in the simulation conducted in python; in (b) the same simulation conducted in ADAMS [219].

The findings for both models are shown below in terms of the vertical movement of three rover-specific points; specifically, Figure 2.50 shows the pin of the left bogie ($\mathbf{B}_l$), the right bogie ones ($\mathbf{B}_r$), as well as the rover center of mass ($\mathbf{G}$) (c-e). The prefixes SA and A in the figures stand for the semi-analytical and ADAMS models, respectively. The displacement, along the vertical direction, of the three sites is specifically reported in Figure 2.50c, Figure 2.50d, and Figure 2.50e for rover speeds of 0.34, 1, and 2 $\mathrm{m\,s^{-1}}$, respectively. Last but not least, Figure 2.50 reports the rover model's findings in terms of roll and pitch angles (f-h).

As measures to assess the disparity between the models, in this example, using ADAMS and SA models, a number of indices are therefore created. Indeed, it has been specifically specified the *Root Mean Square* (RMS) errors between the positions of the points $\boldsymbol{B}_r$ ($RMS_{\mathbf{B}_r|_z}$) and $\boldsymbol{B}_l$ ($RMS_{\mathbf{B}_l|_z}$), called also as the "hips" of the rover, in the SA and ADAMS models. On the other hand, it has been utilized $RMS_{\mathbf{G}|_z}$ for the COM's position, and RMS_χ and RMS_ψ, respectively, for the pitch and roll angles.

Sensitivity analisys

In this paragraph, it is conducted and presented a sensitivity analysis done by changing a selection of variables, including such f and c_q as well as the torque preload $M_{0,\tau_{j,i}}$, in order to determine the impacts of the uncertainties of some semi-analytical design variables, tweak parameters, and improve match with the experimental one.

The Root Mean Square (RMS) error between simulated and experimental curves is specifically chosen as a measure. In order to more accurately replicate the damping phenomena produced in the actual S-Structure, the damping coefficient c_q assigned to the simplified joints in the SA model is the key parameter to tweak. In order to simulate the case of the horizontal drop test, several simulations have been carried out. Every variable that is subject to the above-reported uncertainty is forced to fluctuate randomly in these simulations.

It is ideal that $f \longrightarrow \infty$, according also to Figure 2.26, to more accurately describe the step response characterizing the linear preloaded springs encapsulated inside the generic S-Structure. However, high values assigned to this parameter lead to an increase in discontinuities of the joint dynamics, which has a detrimental effect on the solver performance. The results depicted in Figure 2.52a seek to quantify the impact of this factor. It is noticeable that the computing time t_c required to finish a simulation grows linearly. The same picture also demonstrates that the RMS exhibits a minimum for comparable small values of f before tending to linearly grow. The following factors are likely to contribute to this behavior: the general S-Structure has backlashes; the steel cables are flexible and elastic; and the connection between the cable and the module is accomplished by a loop in the cable. All of these factors work together to cause the preloaded spring response to have a smoother trend than that of the expected step response, which may be best explained by utilizing low values of f.

Figure 2.52b and Figure 2.52c show another noteworthy finding that was produced by changing both c_q and f variables simultaneously. By adjusting both

f and c_q, the response surface characterizing the RMS is shown in the first figure. It is clear that either low values of c_q combined with any of the f ones or low values of f and c_q lead to low values of RMS. Nevertheless, it has been then chosen to utilize a low value of c_q, in this case, equal to $0.14\,\mathrm{N\,m\,s\,rad^{-1}}$, in order to better simulate the step response function of the preloaded linear spring within the S-Structure (with f equal to 1×10^4). Instead, the second figure illustrates the response surface that describes the computing time t_c by changing both f and c_q simultaneously. It is therefore evident that when f and c_q grow, computation time also rises.

Finally, it has been measured the impact of the uncertainty of the estimation of the individual torque preload $M_{0,\tau_{j,i}}$ after selecting the parameters describing the hardness of the step response function f and the joint's torsional damping coefficient c_q. In specifically, a large number of simulations have been performed, with each joint receiving a randomly selected value for the torque falling inside of its normal distribution, that via experimentation has been determined and is listed in Table 2.5. In Figure 2.52d, the outcome of these investigations is shown. More specifically, the graph demonstrates the uncertainty of the left rover pin's vertical position along the z axis (very close to $\boldsymbol{B}_l$) as a result of the preload torques' measurement uncertainty; the figure displays in form of gradient the distribution and includes a 3σ confidence range.

2.3.11 Dynamic models' validation discussions

The outcomes for both the two case studies which have been given in the previous paragraph (see Sec. 2.3.10) are thoroughly explored in this section. Following a summary and discussion of the sources of uncertainty, a thorough examination of the actual data is provided, together with comparisons between the three models.

The rover described in this study is an extremely complex vehicle with several DOFs and hyper-parameters, which necessarily makes the vehicle more complicated overall. This makes modeling and replicating its dynamics when interacting with the surroundings exceedingly challenging. Additionally, there are several sources of uncertainty in the presented rover prototype, including flexibility and elasticity of the structure as well as backlash inside the joints of the generic S-Structure, which together lead the legs to rotate axially during impact. The springs' stiffness and their preload tension values are both subject to a minor amount of error. In addition, the steering hub has shown signs of lability. In fact, when a vehicle impacts the soil's surface, the wheels naturally revolve around the steering axis, indicating the existence of a compliant phenomenon in the steering joint. On the other hand, the wheel hub as well as the rims also exhibit some flexing phenomena. Additionally, the release mechanism which has been utilized to trigger the drop is another source where experimental uncertainty might be detected. In particular, the release mechanism introduces a minor disturbance into the system when it is triggered, which has an impact somewhat on the initial conditions of the rover for the case of drop test situations even though it was planned and manufactured to minimize interference. Figure 2.50a and Figure 2.50b experimental curve's beginning portion provides evidence of this effect.

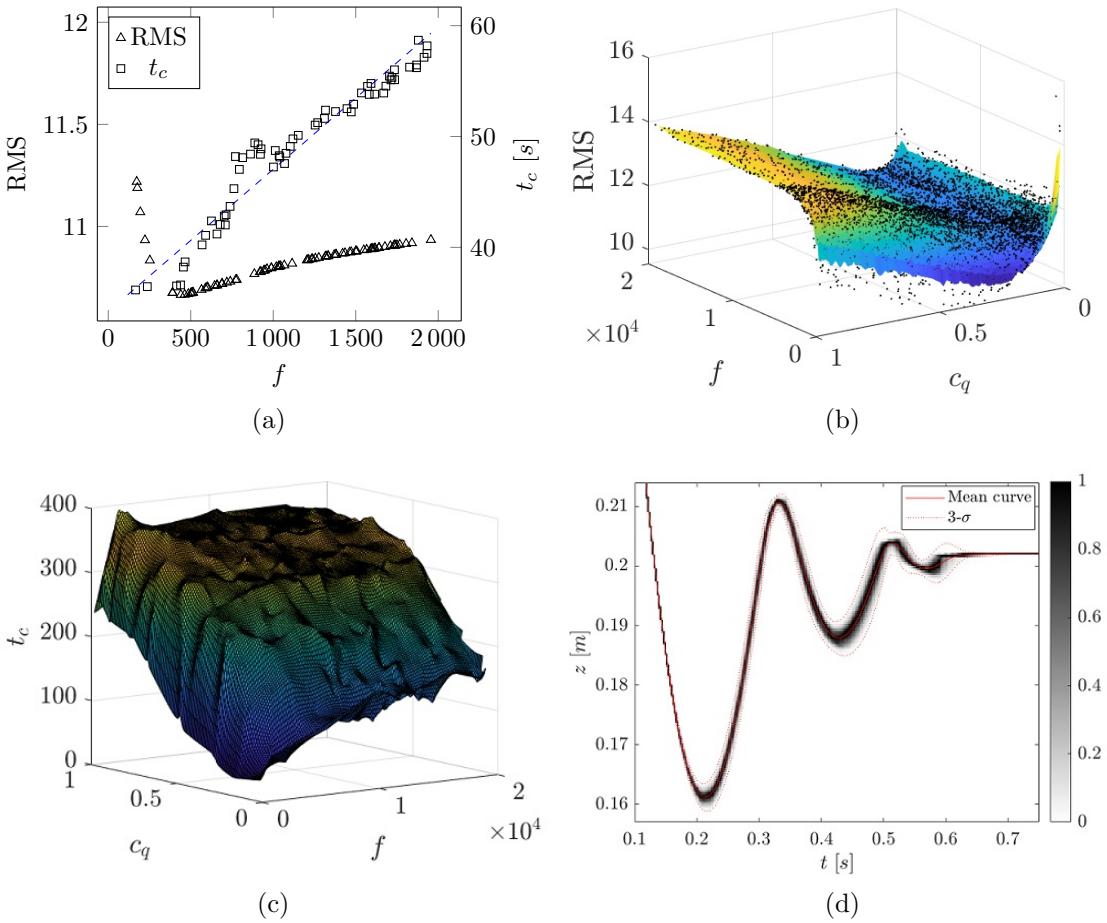


Figure 2.52: In (a) the influence of the parameter f on the computation time t_c in seconds and on the RMS (in mm), together with linear regression for the computation time; in (b) the response surface describing the RMS as the parameters f and c_q (in $\mathrm{N\,m\,s\,rad^{-1}}$) vary; in (c) the response surface describing the computation time t_c; and in (d) influence of the variation of the torque preloads $M_{0,\tau_{j,i}}$ in N m on the vertical displacement of the pin of the left bogie: here, the fuzziness represents a normalized statistical distribution associated with an uncertain value of $M_{0,\tau_{j,i}}$, together with its mean value and 3σ boundaries [219].

It must be stressed that there are several ODE solvers for the numerical models, such as the SA and ADAMS ones. This consideration may also be a factor in the two models' divergence from one another. Lastly, contact with the soil has a significant influence. It has been observed that the wheels bounce rapidly after a collision with the soil; moreover, still, the wheels are subjected to a regular amount of play in the rolling axis because of the relatively loose connection with the drive. When these two effects are combined, the wheel is only slightly affected by soil friction effects during impact and deflection (about from 0.17 to 0.22 seconds in Figure 2.50a), but it is more during the rebounding phase (0.22-0.35 seconds) when the wheels are solidly in touch with the soil. The previously cited play phenomenon is essentially recovered by the bounces, which

means that friction's influence upon the motion during deflection is significantly reduced. In comparison to other curves that are not affected by this effect, the effect results in a larger deflection and a smaller rebound.

The numerical models only partially account for all of the aforementioned factors, which cause some of the models to deviate from the experimental findings. It's noteworthy to notice that the SA model generally exhibits strong adherence following the rebound phase. This is most likely because the damping coefficients c_q assigned to the simplified joints of the S-Structure, which are artificial and have no physical counterparts, were changed to match the model. Throughout the experimental campaign has been observed damping which is likely caused by the complicated behavior of friction phenomena occurring between the wheels and the soil, as explained in the preceding paragraph, and it is only loosely modeled by ADAMS. Throughout contrast, the ADAMS model replicates the S-Structure mechanism, which however has not been fitted including an explicit damping component. Finally, once the motion has stopped, the SA model exhibits amazing coherency with the experimental findings. ADAMS, however, does not adequately represent this. This is most likely because the experimental model exhibits some structural compliance, which the SA model very effectively captures throughout its soft formulation for the elastic response of the preloaded springs, as seen in Eq. (2.13).

Table 2.8: Comparison between semi-analytical (SA) and numerical (Adams) models and experimental (EXP) errors for the two case studies. Drop test: "*at-rest*" error for SA and the Adams models with respect to EXP; deflection error for SA and Adams models with respect to EXP. Drive test: the RMS between the SA and Adams models [219].

Drop Tests			Drive Tests		
$\varepsilon_{r,SA}$	(horizontal)	0.7%	$RMS_{\mathbf{B}_l\|_z}$	($v = 0.34\mathrm{m\,s^{-1}}$)	0.6042mm
	(skewed)	0.2%		($v = 1\mathrm{m\,s^{-1}}$)	0.5525mm
$\varepsilon_{r,Adams}$	(horizontal)	5.1%		($v = 2\mathrm{m\,s^{-1}}$)	0.9335mm
	(skewed)	3.1%	$RMS_{\mathbf{B}_r\|_z}$	($v = 0.34\mathrm{m\,s^{-1}}$)	0.7457mm
$\varepsilon_{d,SA}$	(horizontal)	12.6%		($v = 1\mathrm{m\,s^{-1}}$)	1.1962mm
	(skewed)	30.9%		($v = 2\mathrm{m\,s^{-1}}$)	2.6094mm
$\varepsilon_{d,Adams}$	(horizontal)	7.57%	$RMS_{\mathbf{G}\|_z}$	($v = 0.34\mathrm{m\,s^{-1}}$)	0.4440mm
	(skewed)	24.8%		($v = 1\mathrm{m\,s^{-1}}$)	0.7007mm
				($v = 2\mathrm{m\,s^{-1}}$)	1.5758mm
			RMS_{χ}	($v = 0.34\mathrm{m\,s^{-1}}$)	0.0012rad
				($v = 1\mathrm{m\,s^{-1}}$)	0.0032rad
				($v = 2\mathrm{m\,s^{-1}}$)	0.0035rad
			RMS_{ψ}	($v = 0.34\mathrm{m\,s^{-1}}$)	0.0068rad
				($v = 1\mathrm{m\,s^{-1}}$)	0.0057rad
				($v = 2\mathrm{m\,s^{-1}}$)	0.0072rad

Even with the aforementioned sources of inaccuracies as well as the differences seen in the transitory phase, all the models exhibit excellent conformance in

the resting position if the definitions for both the errors defined in Eq. (2.33) and Eq. (2.34) are considered. The SA and experimental comparisons are good examples of this. In fact, using Table 2.8 as a guide, it is possible to observe that those errors are rather small in the case of a horizontal drop in the resting configuration for the SA model. Considering the error at rest, the numerical model exhibits larger error values, for the skewed drop, with errors as low as $\varepsilon_{r,SA} = 0.2\%$. When the drop is skewed, the greatest maximum deflection errors have been observed to be $\varepsilon_{d,Adams} = 24.8\%$ for the ADAMS model and $\varepsilon_{d,SA} = 30.9\%$ for the SA model. The existence of backlash, friction, and structural compliance are only a few of the factors previously mentioned that contribute to these deflection error values in the skewed scenario. Additionally, there is the challenge of re-creating accurate initial conditions for both the ADAMS and SA models which are consistent with the experimental setup. Last but not least, it should be kept in mind that in the skewed scenario, the measuring plane on the rover is inclined relative to the one of the Motion Amplifier sensor, resulting in accuracy problems in the identified displacements. A bigger deflection error compared to the horizontal drop example is the result of all these elements acting together to introduce inaccuracies.

Figure 2.50(c-h) illustrates the extraordinary consistency observed between the responses of both the two numerical models under consideration, the SA and the ADAMS models, with regard to the drive tests. These findings, nevertheless, need to be taken into consideration in various ways. The angle of pitch between both the models matches exactly, as can be observed in particular from Figure 2.50(f-h), although there are a few discrepancies in the roll angle trajectory. The reason for these variations is that the wheels in the SA model are portrayed as a disk without any depth, but the wheels in the ADAMS model have been modeled as rigid bodies with a finite depth extension. As a result, when the rover encounters the obstacle and attempts to surmount it, it tilts in a rigid way, which causes the contact surface to shift throughout the profile of the wheel as the roll angle grows, this causes also the separation between the steering axis of the wheel and the point of contact of the wheel to increase. This explains why the roll angle which has been observed in the ADAMS model is greater than the SA one even if having equal simulation time.

Since the positions of the three selected points ($\mathbf{B}_l|z$, $\mathbf{B}_r|z$, and the $\mathbf{G}|z$) are dependent on the angle of roll of the rover, the difference seen in this quantity affects all the pins' vertical displacements, as well. In particular, it is a little lower than those represented by the SA model, as can be seen in Figure 2.50(c-e).

A fascinating finding can be observed in the trajectory of the solid curves which exhibit greater frequency oscillations near the peak of their trapezoid shapes, as seen in Figure 2.50(c,d,f,g). This is because the rover is tipped so much that the contact surface is located on the wheels' grousers. In this manner, a hopping-like motion is experienced by the rover as the contact patch switches between one grouser to the subsequent one.

Figure 2.50(d,e,g,h) clearly shows that the SA and ADAMS models' curves have a little amount of temporal shift. They implicitly presuppose that the wheels of the rover generally operate under slip circumstances because of the method

contacts are described. Because of friction, there are differences in the models' points of contact. The magnitudes of slip that are applied to the wheels in the two models differ as a result. This variation obviously affects both the rover's distance traveled and its punctual velocity.

The graphs illustrated in Figure 2.50(c,f) largely follow the geometry of the obstacle for the rover velocity equals to $0.34\,\mathrm{m\,s^{-1}}$, indicating that the effects of the dynamics are relatively minimal. When the rover speed is $1\,\mathrm{m\,s^{-1}}$, as shown in Figure 2.50(d,g), the obstacle's shape can nevertheless be clearly seen, but the dynamics effect can be observed in some places. On the other hand, in the case of $2\,\mathrm{m\,s^{-1}}$, Figure 2.50(e,h), the trajectory is significantly distorted due to the dynamic effects rather than the geometry of the obstruction. The bounces and rebounds of the structure, as well as the intervals during which the wheels do not come into touch with the soil, are easily discernible in these curves.

It has been observed, nevertheless, that consistency between both the numerical models is mostly maintained under control, even during the high-speed scenarios (1 and $2\,\mathrm{m\,s^{-1}}$). Given that the rover's speed throughout the drop impact test is comparable, the two elected speeds for the driving tests are distinguished by their inclusion of significant dynamics effects.

For each simulation example, between the graphs of the ADAMS model and the SA one the RMS is calculated. This has been done in order to offer a quantitative understanding of the conformity between both models. The values again for roll and pitch angles are very low, with observed maximum values about $0.007\,\mathrm{rad}$, and in the instances of the $\mathbf{B}_l|z$ and $\mathbf{B}_r|z$ no greater than $3\,\mathrm{mm}$, as shown in Table 2.8.

2.4 Towing manipulation strategy

In this section, we aim at studying and addressing a very interesting and useful topic in the whole field of mobile robotics, which is that of object manipulation. Indeed, with a view to assigning more and more in number and increasingly varied tasks to mobile robots, the development of systems and methodologies to be able to manipulate objects in the surrounding environment is also being studied. This topic is also of particular interest in the field of mobile robotics for space applications of course. In fact, in addition to the current manipulation techniques used that allow probing, sampling, and collecting samples from the environment, these techniques can be used for other more sophisticated applications of particular interest to the scientific community. Just think of the idea of being able to deploy and use sets of robotic systems at a landing site on a planet, and for these to go and move and manipulate objects in order to build precursor bases prior to human arrival at the site. Specifically, in this section, a manipulation technique based on the towing, by means of elastic elements, of objects placed on the ground will first be studied in detail and formally, in a general way. Next, the focus will be shifted to the development of a path planner for the towing robot, which is intended to generate trajectories that the latter must follow in order for the object to be towed from a starting point to an endpoint and to follow a certain predetermined trajectory. This study also draws attention to the use and validation of this methodology for the Archimede rover, which was described in the previous section. In addition, the influence that constraints on the rover's steering joints have on the generated trajectories is quantified.

It is evident that the main interaction that needs to be carefully resolved in this study is the interaction that the robot has with the towed object. This, unlike the interactions seen in the previous sections, is an active type of interaction. In fact, the rover, through the use of cables for towing operations, generates interactions that allow it to actively manipulate objects placed on the ground. Finally, this interaction falls into the macro group of environment-robot interactions. In this study, this is not the only interaction that the mobile robot is subject to, but as seen in the previous section the rover is subject to interactions with the ground through the wheels. However, the latter is not very relevant for the purpose of this study. In fact, particular attention in this study is rather placed on the interaction that occurs between the object that is to be towed and the ground. This as will be seen is of fundamental importance in order to develop a proper methodology for object towing and the development of a path planner for the robot.

Summarizing, the contributions of this section can be listed in:

- The formulation and application of a quasi-static approach for the path planning of a towed object;
- The implementation of a continuous contact model and the development of two different dynamics models for the towing problem;
- The application of the approach to the Archimede planetary rover prototype

which is subject to joint limits constraints, and successive determination of the influence of its kinematics;

- The experimental validation of the proposed manipulation approach.

2.4.1 Introduction

In the space exploration field, planetary rovers have long been the focus of interest and research. It is becoming increasingly obvious and necessary to focus research on how such robotic systems, which usually possess a certain degree of autonomy, can be used in order to prepare and therefore set up the colonies and, consequently, the manipulation and movement of objects, given the ever-increasing interest in space exploration and, in particular, potential and in a relatively close future human settlements on the Moon. There are several instances of rovers that can manipulate things and collect samples [229–231]. The Mars 2020 Perseverance Rover, which does have a robotic arm with five DOFs equipped to be used in collecting and storing samples, is the most current and impressive example [232, 233]. Therefore, it is expected that the usage of planetary rovers for handling, transport, and payload manipulation for the goal of setting up human settlements would soon become a reality. The problem, despite being of great interest to the scientific community, poses two main challenges: the first regards the planning operations, or how to plan the rover's motions such that the object is correctly placed in the intended position [234, 235]; the second, instead, regards the safety aspects, or how to position the object safely while avoiding the environment's obstacles.

There is a significant amount of literature on various types of object manipulation techniques used by mobile robots. Furthermore, a large number of robot collaboration strategies include working together to accomplish the given common goal. For instance, flexible components tied all around the object required to be manipulated have been explored for manipulation utilizing prehensile [236, 237], and non-prehensile [238] strategies. By using sticks and strings Yamashita et al. studied cooperative manipulation of objects, where these elements have been utilized as tools for pushing or pulling, respectively, in 1998 [239]. Other transportation methods are described in the literature [240–242]; they typically focus on control and formation control issues and include goods delivered on top of a mobile robots team. A similar strategy that has been explored by Hichri et al.[243] includes the potential for many robots to collaboratively push the object first before lifting and then transporting it, in 2014. Pushing is an extensively studied manipulation technique, which can be either be carried out by a single mobile robot pushing the object [244–246] or it is carried out cooperatively [247], which may also involve caging [248]. In 2014 Ohashi et al. contributed with an intriguing study in which they adopted a mixed approach. This entails tilting the object to allow handcarts to fit underneath it, then pushing the object to complete the conveyance [249, 250].

Towing is indeed a manipulation technique that often uses cables; it has received much study and finds utility in practical situations in the realms of maritime and airborne applications. Towing is employed in the former when "escort

tugs" are deployed to assist huge ships [251, 252], or to pull tiny submarines [253, 254]. The latter, however, uses towing mostly to convey hanging payloads [255–259]; nevertheless, a towed decoy system to shield military aircraft from radar-guided missiles has also been investigated [260].

While towing is frequently employed in the maritime and aerial industries, there aren't many studies that apply it to mobile robots which operate on the ground. Since cables add a significant amount of complexity, other sorts of techniques, such as pushing, are favored for the latter, as was already previously indicated. Two rovers connected by a tether are seen in the Axel rover [235, 261], while Cliff-bot [262] uses a similar strategy. A quasi-static (QS) method was employed in order to tow an object having a "L" shape featuring three point-like points of contact in the study presented by Cheng et al. in the challenge of towing objects using several collaborating mobile robots. In their study, each of the mobile robots is attached to the object [263]. Passive tails attached to the backside of mobile robots were proposed by Kim et al. in order to hook and afterward tow items [264, 265]. On the basis of ant behavior, Wilson et al. created models for object towing in 2018 [266].

It is known as the "planar sliding issue" the problem that rise when an object that has been placed on a flat surface is forced to slide as a result of pushing or towing operations. When two bodies are sliding relative to one another, friction exists at the contact surface. When the contact interface between two bodies cannot be assumed to be a single point of contact and is instead a surface, in addition to the conventional forces of friction, also a frictional moment takes place as well. The modeling of an item moving on a two-dimensional surface, however, is not simple. Additionally, the motion of the sliding item is largely dependent on the distribution of pressure, which generates between the body and the contact surface, and is typically unpredictable and cannot be predicted beforehand. Another source of uncertainty is the distribution of friction between both sliding elements; in reality, friction behavior cannot be universally assumed to be constant and uniform throughout the whole surface and may vary depending on the region. Additionally, the characteristics of friction may also change over time. Furthermore, the pressure distribution is influenced by the external loads that are applied to the sliding item.

The terrain is often soft and uneven while working with planetary rovers [267, 268], as opposed to hard and smooth. Many works have explored the interaction phenomena which occur between wheels and soil [269, 270], and even in some cases with emphasis on experimental [271–273].

Numerous researchers have concentrated their efforts and research on what is referred to be the field of the "mechanics of sliding objects". Mason [274], who developed an essential theory that specifies the sense of rotation (clockwise or counterclockwise) of a pushed item regardless of the pressure distribution, put out a highly remarkable study; Goyal et al. [275, 276] proposed the idea of a friction limit surface which has the role of relating the net frictional load, which is represented by friction force and friction moment, to the motion of an item sliding; Howe and Cutkosky suggested that an ellipsoid may be used to approximate the friction limit surface introduced by Goyal et al. and therefore

create an exact and quantifiable relationship between both the net frictional load and the object sliding speeds [277]. Later, these elements were utilized for robotic pushing [278], stable robotic pushing [279], robotic pushing while anisotropic friction characteristics were unknown [280], and robotic cooperative towing [263]. The same idea has been widely applied to robotic soft-fingers, which typically include pressing an item on a surface and subsequently moving it with the help of an external agent, as an instance, a robotic arm [281–286]. Models in this domain are simpler since the pressure distribution is often assumed to be known in advance.

The current effort focuses on the capacity to tow, by using a cable, an object of general shape. At this point in the research process, cables are thought of as massless elastic constraints that solely exhibit traction behavior. In general, manipulation by towing enables manipulation of items that cannot be lifted, for example, due to their weight or size. Furthermore, it is thought that a system utilizing a strategy, which employs towing manipulation, could be preferred over the traditional pick-and-place operations that are currently used on planetary rovers. The latter would necessitate the rover being equipped with the robotic arm, which is typically a heavy apparatus, making the overall robotic system way harder to design given the typical strict weight constraints for space missions. The operating life of the rover is further impacted by additional load due to increased wear, power usage, risk of embedding in loose soil, etc. The separation of the payload which comes with towing may theoretically be a viable choice in extreme situations.

The study discussed in this section is related to the work on cooperative towing given by Cheng et al. [263]. The two studies are, nevertheless, distinct in a number of ways. In order to calculate the pressure distribution in their research, the authors first employ a three-point contact model; in contrast, in this study, it is assumed a bi-dimensional continuous contact surface, and that is incompatible with the assumption of point contact. As the towing actor, it has also been taken into account the Archimede planetary rover prototype. As it has seen in the previous section, this rover is subject to more complicated kinematics restrictions, such as steerable wheels and constrained steering joints. This fact causes the mobility of the mobile robot to decrease. Finally, Cheng's method employs a fleet of three robots, whereas in this case, it has been used just one rover to plan the position of the towed object. A sole mobile robot towing the item makes the multi-robot system less manageable and the planning more difficult, even if it is helpful to utilize several robots to manipulate an object since it theoretically permits the object to take any position. A space research mission employing a multi-robot system might be too costly and difficult to operate; on the other hand, using a single robot simulates the potential circumstance where a planetary rover is the sole towing agent available.

Finally, this section is structured as follows: in Sec. 2.4.2 is outlined the problem of towing an object. Subsequently, it is presented the contact model adopted, the development, and a detailed description of the quasi-static model used for addressing the problem and generating the trajectories. Moreover, it will be presented the two different dynamic models for the object being towed as well

is presented the model of the rover model subjected to joint limits constraints and presented a path planner which leverages the concept of the ICR projection and the orientation look-ahead. In Sec. 2.4.3 are reported the findings obtained for the selected simulated use cases, first regarding the path planning, while secondly for the sensitivity analysis performed in order to capture the influence that some hyper-parameters have on the presented approach. Finally, in Sec. 2.4.4 are reported the experimental results that have been obtained for the validation of the approach.

2.4.2 Methodology

The major goal of this study is to determine how an item should be towed in order to follow a specified reference trajectory, specifically what the towing agent's trajectory should be in order for the object to actually follow the reference trajectory. In this study, it is assumed that perhaps the motion of the towed item is suitably slow, thus allowing to ignore the inertial components and it has been elected to derive the solution by using the quasi-static approach.

In reality, the towing agent's trajectory—or, to put it more broadly, its motion—is generated as a result of this quasi-static, QS in short, approach. Due to some established assumptions, such as inertial effects, it must be confirmed that these do not significantly deviate from the desired nominal route in order to verify the technique. Due to this, it has been gone through two dynamic models in the paragraphs that follow: one that is only dependent on the forces applied to the towed object as determined by the quasi-static technique, and another that explicitly takes into account the towing agent's trajectory. By correlating the nominal trajectory of the towed item with the calculated real trajectories using the dynamic models, it is therefore possible to validate the methodology by using those models.

Thinking about the scenario where a generic-shaped body is being towed by an external force while sliding on a horizontal plane. The extremely broad case is illustrated in Figure 2.53, where it is assumed that the body is bound to travel along a reference trajectory $\boldsymbol{s}_o(t)$ having an arbitrary shape. In this context, the triple (x, y, ϑ) specifies the state of the object, which is referred to as a generic point $\boldsymbol{P}$, represented in the inertial frame of reference $(\boldsymbol{O}, \hat{\boldsymbol{e}}_x, \hat{\boldsymbol{e}}_y, \hat{\boldsymbol{e}}_z)$. Furthermore, $\boldsymbol{P}$ is considered to be the point in this context that must follow the trajectory in this manner. Let's introduce then $(\boldsymbol{P}, \hat{\boldsymbol{\imath}}_o, \hat{\boldsymbol{\jmath}}_o, \hat{\boldsymbol{k}}_o)$ as the reference frame fixed to the object. The applied towing force $\boldsymbol{T}$ to the anchor point $\boldsymbol{Q_T}$, the frictional force $\boldsymbol{F}_k$ which is applied to the center of friction $\boldsymbol{Q}_k$, which is the result of a friction coefficient uneven distribution throughout the contact patch, as well as the applied frictional momentum $\boldsymbol{M}_k$ all together affect the object's velocity, indicated with $\boldsymbol{v}$.

Specifically, Figure 2.53a depicts the standard free-body diagram of a body being towed and sliding on a planar frictional terrain, whereas the relation between both the object and rover trajectories, represented through an elastic unilateral tether having a length at rest l_0 and stiffness κ, is shown in Figure 2.53b.

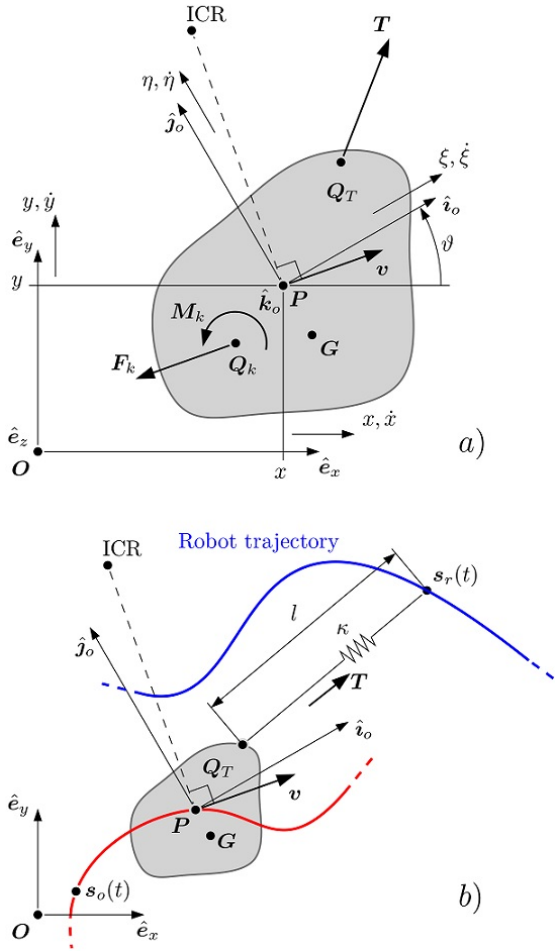


Figure 2.53: General diagram regarding the motion of an object along a trajectory, over a frictional surface, and towed by a rover: in (a) the free-body diagram of the towed object; in (b) focus on the trajectories of the object and of the rover.

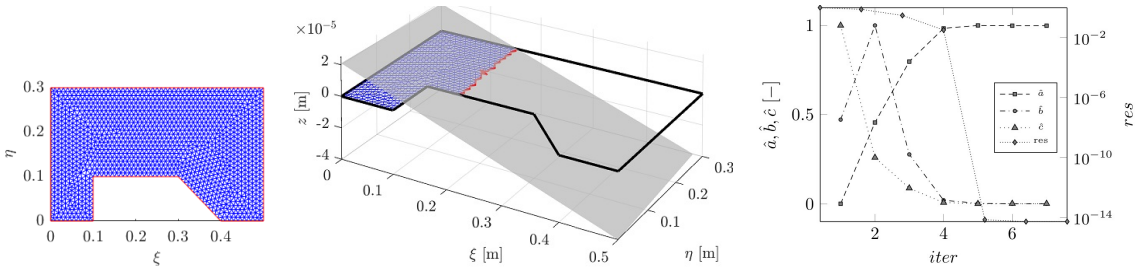
Contact modeling

The method used to create the geometry of the object as well as the modeling of the two fundamental forces, that are exchanged between the object and the horizontal frictional surface (also known as pavement, for short) that makes up the contact interface, are both described in this paragraph. Specifically, these two forces are the normal and tangential contact forces to the contact surface. The first is determined from the knowledge of the distribution of the contact pressure; while the second is the frictional force, determined from the knowledge of the Coulomb friction model's parameters.

- *Contact Patch Geometry Definition*: A CAD software package is employed in order to model the object's shape, precisely its contact surface. First, a STL file format is exported and subsequently loaded into the simulator which has been developed in MATLAB[8]. The mesh needed to evaluate the

[8]The data and the source code are publicly available at the following GitHub repository: `https://github.com/matteocaruso1993/towing-rover.git`.

pressure distribution all along the contact patch is eventually generated by the simulator. An imbalanced cube (squared contact surface), which will be primarily utilized in the sensitivity analysis, which will be presented in the later sections, and a generic geometry with the contact surface, as illustrated in Figure 2.54a, are the two primary geometries utilized within this study.



(a) (b) (c)

Figure 2.54: Contact model. In (a) the mesh representing the initial contact surface; in (b) the mesh of the effective contact surface for the case of an extremely unbalanced object towards the left side and focus on the cutting plane, shown in grey; in (c) the plot showing the convergence of the normalized plane coefficients and the residuals.

- *Normal Force*: The distribution of the contact pressure, which again is unknown a *priori* and typically is not uniform, is used to calculate the resulting normal contact force. The contact pressure distribution that a body experiences, while it is lying on a surface, relies on the object's mass distribution, unbalanced loads, and external forces acting on it. Typically, when two bodies are squeezed together, the pressure of the contact results in a change of both their geometry, which may then result in a mutation of the effective portion of the region where contact occurs. This is demonstrated in Figure 2.54b, in which the contact patch, or the region where contact takes place, is in fact less than the object's complete extent, which instead is depicted in Figure 2.54a. As a result, the pavement is represented as a continuous sheet of linear springs with stiffness k_{gr} in order to comprehend the pressure distribution. Additionally, if the body only slightly penetrates the pavement, it is mathematically viable to represent the distribution of both the penetration and consequent pressure of the object as a plane. For simplicity's sake, the plane is defined as follows within the body fixed frame:

$$z(\xi, \eta) = a\xi + b\eta + c \tag{2.35}$$

where the values ξ and η are, respectively, the increments along the versors of the fixed reference frame of the object $\hat{\boldsymbol{\imath}}_o$ and $\hat{\boldsymbol{\jmath}}_o$; whereas a, b, and c are the three coefficients which describe the plane in the three-dimensional space.

Considering Eq. (2.35), sheet k_{gr}'s stiffness, together with the observation that pavements can transmit normal forces when compressed only, it is straightforward to formulate the following rule for the distribution of the contact pressure distribution $p(\xi, \eta)$, and therefore the normal force:

$$p(\xi,\eta) = \begin{cases} -k_{gr}(a\xi + b\eta + c), & z \leq 0 \\ 0, & z > 0 \end{cases} \quad \rightarrow \quad F_n = \int_S p(\xi,\eta) dS, \tag{2.36}$$

where F_n represents the normal contact force that is exerted at the center of pressure and S represents the integration domain, which matches with the contact patch surface.

In this study, a *cell-centered Finite Volume Method* (FVM) has been used rather than a closed-form solution in order to accomplish integration over general domains. This has been done since the integration domain, or the contact patch could have any shape. Specifically, in this study when a surface integration is required, such as for the distribution of $p(\xi, \eta)$, the FVM will be applied. This also holds true for the components of the frictional force $\boldsymbol{F}_f$ and $\boldsymbol{M}_f$, as will be demonstrated in the later paragraphs.

- *Friction Model*: As done in Sec. 2.3.1 the model proposed by Makkar et al. in [220], is leveraged to describe the friction coefficient between two sliding bodies as a continuous and always differentiable function as done in Sec. 2.3.4, and whose formulation, for comfort purpose, will be reported here again as well,

$$\mu(\boldsymbol{v}) = \gamma_1 \left(\tanh\left(\gamma_2 \boldsymbol{v}\right) - \tanh\left(\gamma_3 \boldsymbol{v}\right)\right) + \gamma_4 \tanh\left(\gamma_5 \boldsymbol{v}\right) + \gamma_6 \boldsymbol{v} \tag{2.37}$$

where $\boldsymbol{v}$ is the relative velocity of the sliding bodies and the γ_i terms again are the positive constants that specify the friction model. For simulation stability and the creation of high-performance continuous controllers, both the continuity and the differentiability of the function describing the friction are crucial [220]. Since it is not really practical to express the γ_i parameters explicitly, as done as well in section Sec. 2.3.1, it has been chosen more useful engineering parameters like μ_s, μ_k, v_s, and v_k, which again stand for the static and dynamic friction coefficients, stiction, and friction velocities, respectively. The friction model was then created by computing the γ_i parameters using also in this case a Dual Annealing Algorithm [287].

The coefficient of friction μ, in general, is a characteristic that may take on different values over the contact patch; as a result, a distribution of values $\mu(\xi, \eta, \boldsymbol{v})$ can be used to characterize it. The friction force is therefore defined as follows according to [274],

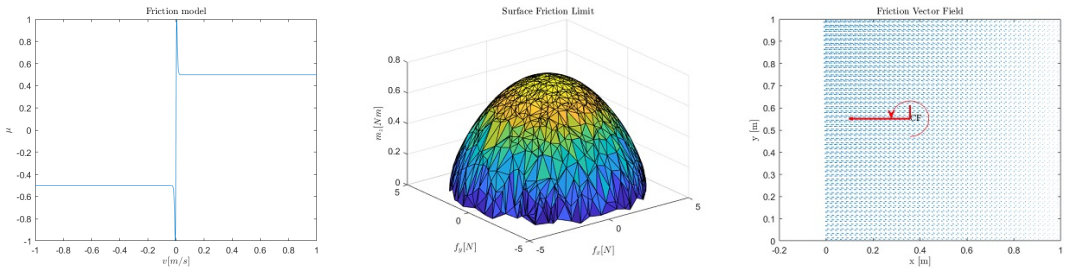
$$\boldsymbol{F}_f = -\int_S \mu(\xi,\eta,\boldsymbol{v}) p(\xi,\eta) \frac{\boldsymbol{v}(\xi,\eta)}{||\boldsymbol{v}(\xi,\eta)||} dS, \tag{2.38}$$

while the moment due to frictional forces is then defined as follows,

$$\boldsymbol{M}_f = \int_S -\boldsymbol{r}(\xi,\eta) \times \mu(\xi,\eta,\boldsymbol{v}) p(\xi,\eta) \frac{\boldsymbol{v}(\xi,\eta)}{||\boldsymbol{v}(\xi,\eta)||} dS, \tag{2.39}$$

in which the position vector of a general point which is represented by the pair (ξ,η) and in relation to the frictional center $\boldsymbol{Q}_k$ is denoted with $\boldsymbol{r}(\xi,\eta)$. The notion of the *Friction Ellipsoid* relates the frictional forces and the friction moment, an illustration of it can be seen in Figure 2.55b; for a thorough discussion, we refer the reader to the work of Goyal et al. [276]. The state of the friction, intended as a vector defined with the components $[F_{f,x},\ F_{f,y},\ M_{f,z}]$, must lie inside this ellipsoid. The region inside the surface determines the states at which there is no motion, while the surface determines the limit at which the object is in motion.

Figure 2.55a shows the synthesized friction coefficient, using the model proposed by Makkar et al. as a function of the sliding speed. In this figure can be identified the static friction coefficient, the Stribeck effect, and the dynamic friction coefficient. In Figure 2.55b is shown the friction ellipsoid obtained for the cube case confirming the concept introduced by Goyal et al. Finally, in Figure 2.55c is reported the friction force field acting on the cube under a sample motion.



(a) (b) (c)

Figure 2.55: Friction properties for the case of the block with a side length of 1 m: in (a) the friction coefficient trajectory as a function of the sliding speed; in (b) the obtained friction ellipsoid and in (c) the friction force field the contact patch experience.

Quasi-Static Model (QSM)

According to the preceding paragraph, Eq. (2.35) and Eq. (2.36) represent the object's penetration and the distribution of contact pressure, respectively, whereas Eq. (2.38) and Eq. (2.39) describe the frictional force as well as the frictional momentum vectors, respectively. However it is still unclear at this point how it is possible to calculate the object penetration (and consequently the pressure distribution). In order to calculate the pressure distribution, the *Quasi-Static Model*

(QSM), which disregards the influence of inertial forces, is described within this paragraph.

There are three equations that must be solved in order to determine the previously introduced coefficients required for defining the plane (a, b, and c) in the space. As a result, it has been taken into account the static equilibrium condition along $\hat{\boldsymbol{k}}_o$ and the moment equilibrium condition around $\hat{\boldsymbol{\imath}}_o$ and $\hat{\boldsymbol{\jmath}}_o$ having the pole for its computing lying on the contact patch. Hence, the following can be used to express these equilibrium equations for the QSM:

$$\sum_{i=1}^{N} \boldsymbol{F}_{ext,i} \cdot \hat{\boldsymbol{k}}_o + \int_S p(\xi,\eta) dS = 0 \tag{2.40}$$

$$\left(\sum_{j=1}^{M} \boldsymbol{M}_{ext,j} + \sum_{i=1}^{N} (\boldsymbol{P}_i - \boldsymbol{\Pi}) \times \boldsymbol{F}_{ext,i} \right) \cdot \hat{\boldsymbol{\jmath}}_o - \int_S \xi p(\xi,\eta) dS = 0 \tag{2.41}$$

$$\left(\sum_{j=1}^{M} \boldsymbol{M}_{ext,j} + \sum_{i=1}^{N} (\boldsymbol{P}_i - \boldsymbol{\Pi}) \times \boldsymbol{F}_{ext,i} \right) \cdot \hat{\boldsymbol{\imath}}_o + \int_S \eta p(\xi,\eta) dS = 0 \tag{2.42}$$

where $\boldsymbol{F}_{ext,i}$ represent the i-th vector of the external force applied to the object and having application point $\boldsymbol{P_i}$ with $i = 1 \dots N$, $\boldsymbol{M}_{ext,j}$ is the j-th vector of the external moment applied to the body with $j = 1 \dots M$, and $\boldsymbol{\Pi}$ represent a generic pole with respect to which the momentums are calculated. The system above may also be written as a $\boldsymbol{A_p x_p} = \boldsymbol{B_p}$ linear system with $\boldsymbol{x_p} = [a, b, c]$ by replacing the pressure distribution law; it can be demonstrated that the matrix $\boldsymbol{A_p}$ only depends on the shape of the contact surface as well as the "stiffness" of the surface, whilst the matrix $\boldsymbol{B_p}$ solely depends on the external forces and momentums. The subscript p is used to differentiate these terms from the ones used in the previous sections. The closed-form solution of the linear system makes it straightforward to define the plane. However, as previously mentioned, the FVM is used to compute the surface integral terms.

The pressure distribution is influenced by external forces, as indicated in the preceding paragraphs and demonstrated by the equations above. As a result, the pressure distribution must constantly be recalculated. The following hypotheses, which let computing the pressure distribution just once, at initialization, were included to avoid this problem:

- The towing force is parallel to the contact patch;
- There is zero vertical offset of the towing force, with respect to the contact plane.

It needs to be noted that even once the plane is identified, it may not be the appropriate one. This is so that it is clear why the pavement only responds forcefully under compression but not in traction, as was previously indicated. It is improbable that a localized area of the ground would be subjected to traction loads whenever an eccentric force is exerted. In reality, just a small contact surface experiences a ground response pressure under extremely eccentric vertical pressures. This being said, the plane has to be determined through an iterative procedure in order to solve this issue. At every iteration, the portion of the

support surface that is affected by traction is deducted from the original support surface S. As a result, a new support surface S' and new mesh are produced. This procedure continues only until the residuals, denoted by the symbol res, reaches the convergence, where the residuals are defined as $res = ||\boldsymbol{A'_p x_p} - \boldsymbol{B_p}||$, where $\boldsymbol{A'_p}$ corresponds to the matrix $\boldsymbol{A_p}$ but calculated on S'. Figure 2.54 depicts the procedure of removing the contact patch recursively for calculation, with Figure 2.54a and Figure 2.54b depicting the initial original mesh as well as the reduced mesh as a result upon convergence, respectively. The convergence of both the residuals and the normalized plane coefficients can be seen in Figure 2.54c, in contrast. The example of an extremely eccentric vertical stress acting on a generally shaped item is the subject of these early findings.

Additionally, the just-explained process of iteratively removing integration domain portions, computation of the plane coefficients, and its implementation is reported as pseudocode in Alg. 1.

Algorithm 1 Pseudocode for the computation of the pressure distribution along the contact patch

$tol \leftarrow 1e^{-6}, \quad res \leftarrow 1e^{6}$
$M \leftarrow loadModelFromSTL(path)$
$S \leftarrow getMesh(M)$
while $res > tol$ **do**
  $(A_p, B_p) \leftarrow getABMatrix(S, F)$
  $(a, b, c) \leftarrow getPlaneCoefficients(A_p, B_p)$
  $(z, p) \leftarrow computePlaneAndPressureField(a, b, c)$
  $S' = S(z \leq 0)$
  $(A'_p, B) = getABMatrix(S', F)$
  $res = ||A'_p x_p - B_p||$
  $S = S'$
end while

It is possible to calculate the frictional force and moment vectors after identifying the appropriate plane and, consequently, the contact pressure distribution. The needed towing force $\boldsymbol{T}$, required for the point $\boldsymbol{P}$ to trace the provided trajectory $\boldsymbol{s}_o$ thus, may finally be computed by using the QSM technique once more to solve the equations for the translational components along the $\hat{\boldsymbol{\imath}}_o$ and $\hat{\boldsymbol{\jmath}}_o$ axis. It's important to note that modeling the contact surface using a mesh enables the description of any shape of the object. In addition, when combined with modeling the friction coefficient with Eq. (2.37), it enables the simple implementation of a general Coulomb friction model, which may then account for anisotropic friction, or $\mu(\xi, \eta, v)$.

Dynamics model

A bi-dimensional dynamics model has been developed for the sliding motion of a towed body over a flat surface in addition to the previously disclosed quasi-static model. Specifically, it has been chosen to develop it in order to compare the outcomes of the QSM technique as well as the outcomes of the experimental testing

and the numerical simulations. With three DOFs, a generic rigid body bound to slide over a flat surface can be totally represented by the vector of the generalized coordinates $q = [x, y, \theta]$ and expressed in the inertial frame. Therefore, just three second-order ODEs are required, as shown in the following. By using the Newton method, and with reference to Figure 2.53, we obtain:

$$m_o \ddot{x} = \sum_i \boldsymbol{F}_i \cdot \hat{e}_x \tag{2.43}$$

$$m_o \ddot{y} = \sum_i \boldsymbol{F}_i \cdot \hat{e}_y \tag{2.44}$$

$$I_{zz} \ddot{\vartheta} = \sum_i (\boldsymbol{P}_i - \boldsymbol{\Pi}) \times \boldsymbol{F}_i + \sum_j \boldsymbol{M}_j \cdot \hat{e}_z \tag{2.45}$$

where m_o represents the mass of the object being towed, $\boldsymbol{\Pi}$ represents an arbitrary point chosen for the calculus of the momentums, I_{zz} is referred to as the inertia component along the body's axis z, $\boldsymbol{P}_i$ represent the point of application associated to the $i-th$ force $\boldsymbol{F}_i$; while $\boldsymbol{M}_j$ represent the j-th applied moment. Moreover, as outlined in Sec. 2.3.4 for the dynamics model of the rover, it is possible to define an additional vector $\boldsymbol{u} = \dot{\boldsymbol{q}}$, in order to represent the system in a state space form. It follows that the augmented vector is $\dot{\boldsymbol{x}} = [\dot{\boldsymbol{q}}, \dot{\boldsymbol{u}}]$, hence the resulting augmented system has 6 first-order ODEs. The system can be also seen in the most general formulation $A\dot{x} = B_{rhs}$ and therefore be solved with any integration scheme. Specifically, within this investigation, it has been elected to use a solver for stiff equations, namely *ode15s* [288].

In relation to the towed sliding dynamics simulations two primary instances are taken into consideration. Nevertheless, for both instances, the quasi-static method is required as a first step necessary in order to determine the force trajectory of the towing force $\boldsymbol{T}(t)$, which is in turn necessary for the object to follow the designated trajectory $\boldsymbol{s}_o$. The aforementioned two cases are:

- *Case 1. FeedForward Dynamics (FFD):* Using spline curves to interpolate the towing force trajectory $\boldsymbol{T}(t)$, which has been acquired via QSM, the towing force has been therefore applied directly to the object, as an applied external force;

- *Case 2. FeedBack Dynamics (FBD):* The rover trajectory $\boldsymbol{s}_r(t)$ is first calculated using the towing force trajectory $\boldsymbol{T}(t)$ given from QSM and is then interpolated using again spline curves. The dynamics simulations are then conducted using the trajectory $\boldsymbol{s}_r(t)$ rather than the force trajectory, and thus the towing force is calculated by treating the tether as an only traction spring as follows:

$$\boldsymbol{T} = \begin{cases} \kappa \left(l_0 - ||\boldsymbol{P}_r - \boldsymbol{Q}_r|| \right) \frac{\boldsymbol{P_r} - \boldsymbol{Q_r}}{||\boldsymbol{P_r} - \boldsymbol{Q_r}||}, & ||\boldsymbol{P_r} - \boldsymbol{Q_r}|| \geq l_0 \\ \boldsymbol{0_v}, & ||\boldsymbol{P_r} - \boldsymbol{Q_r}|| < l_0 \end{cases} \tag{2.46}$$

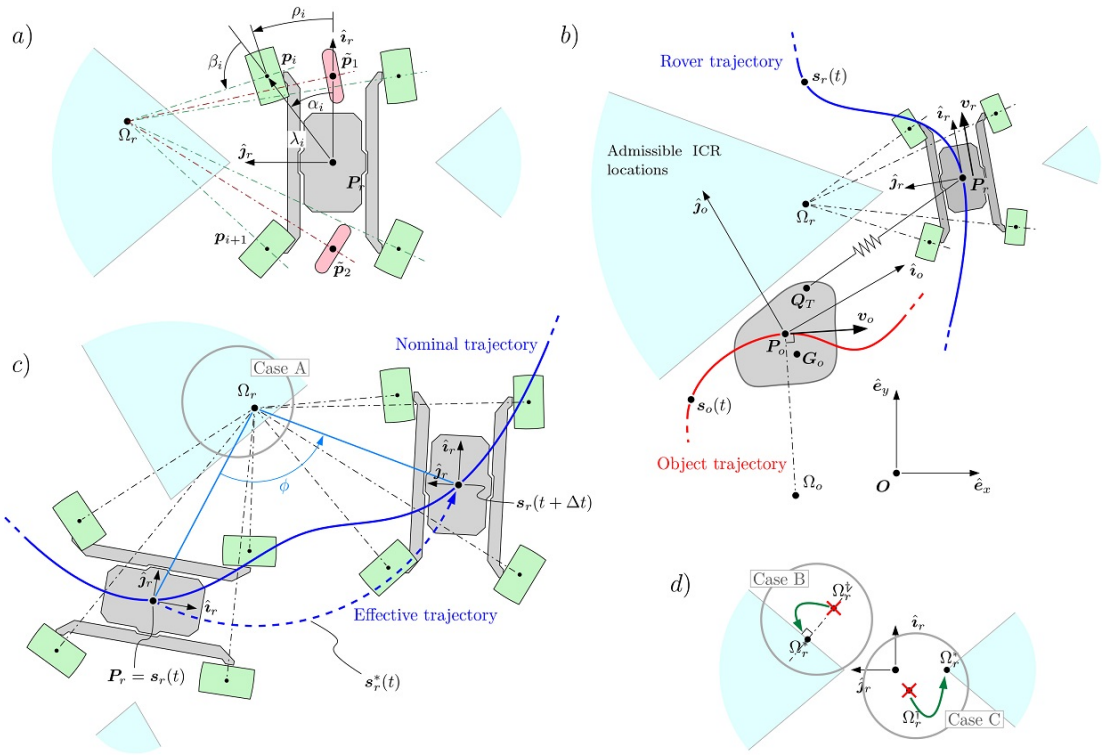


Figure 2.56: Kinematics model of the rover and path-planning definitions. In a) the model's main geometrical parameters are shown, highlighting the reduced 2-wheel bicycle model (red) equivalent to the full 4-wheel rover (green); in b) the relations are shown between the towed object and the rover, with emphasis on the respective trajectories $\boldsymbol{s}_o(t)$ and $\boldsymbol{s}_r(t)$. Both in a) and b) the admissible ICR location area is shown in cyan color. In c) the nominal trajectory is shown and compared to the computed "effective" trajectory approximation around the ICR $\boldsymbol{\Omega}_r$; in d) the cases are shown (Case B and C) where the computed ICR is not feasible.

Rover model

As already mentioned to the planetary rover prototype Archimede, extensively described in Sec. 2.3, is assigned the duty of towing the object. Specifically, the QSM approach described in this paragraph allows for the synthesizing of the nominal rover trajectory, which we will denote with $\boldsymbol{s}_r(t)$, starting from the knowledge of the object's planned trajectory $\boldsymbol{s}_o(t)$. Nevertheless, the actual trajectory invariably deviates from the nominal one due to the kinematics and dynamics of the rover. Additionally, it has been provided a non-holonomic model for the Archimede rover that takes into account the total kinematics, including in particular the steering angle restrictions for each wheel, in order to assess this disparity. Deliberately, the rover's dynamics is omitted within this study.

1. **Kinematics Model**: For this purpose, it has been used the definitions and the geometry that are shown in Figure 2.53b and in Figure 2.56. While the basic geometry of the rover is specified in Figure 2.56a, Figure 2.56b illustrates the connection and interaction between both the rover and the

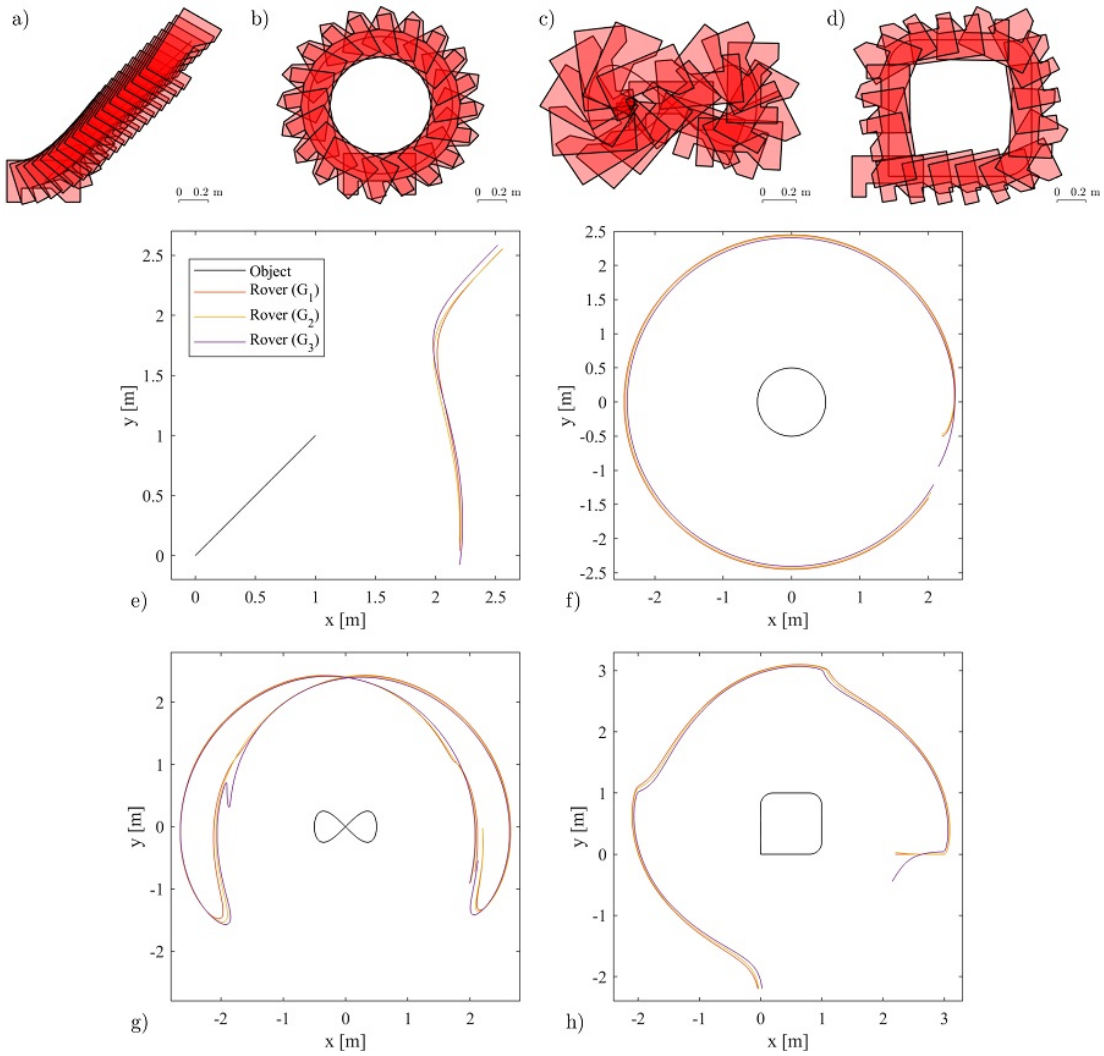


Figure 2.57: Results of the QSM, under the assumption that the motion of the object happens so slowly that the assumption of quasi-staticity of the motion is valid. In the first row, the motion of the object is shown when obtained through QSM for the line (a), the circle (b), the lemniscate (c), and the square (d) trajectories cases. In the second and third rows, the nominal trajectories are reported of the object (black) and the nominal trajectories of the towing rover. Moreover, it shows the effect that the unbalancing of the object, by imposing the gravity application point in G_1, G_2, and G_3, has on the nominal trajectories of the towing rover: for the line trajectory (e), the circle trajectory (f), the lemniscate trajectory (g) e the rounded corner square trajectory (h).

towed object. As introduced in the previous paragraph regarding the QSM method: by considering stiffness κ and length l_0 at rest, the rover's connection to the towed object is described as a unilateral elastic constraint (traction only). Additionally, the steering is controlled by the rover control system in such a way that the point $\boldsymbol{P}_r$ of the rover tracks the trajectory $\boldsymbol{s}_r(t)$, allowing the item to move through the introduced elastic relation and it is able then to follow its own predetermined trajectory $\boldsymbol{s}_o(t)$.

In this investigation, the kinematics of the four-wheel steering rover Archimede is obtained with a different, but analogous method, with respect to the ones presented in Sec. 2.3.5. This method is based on the fact that the rover can be thought of as a bicycle system. In fact, as can be seen in Figure 2.56, non-holonomic four-wheel kinematics, like that of the rover, may be compared to a bicycle with two steerable wheels. The same image shows that each one of the rover's i-th wheel may be characterized by the parameters α_i λ_i and β_i. Additionally, the steering angles ρ_i are subjected to the relation: $\rho_i = \alpha_i + \beta_i - \pi/2$. The generalized coordinates vector of the system, expressed in the inertial reference frame, is instead defined as $\boldsymbol{q_r} = [x_r, y_r, \vartheta_r]^T$, where the subscript r is in this context used to differentiate the generalized coordinate vector from the one used for the object. Moreover, the rotation of the wheels vector is designated by $\boldsymbol{\varphi} = [\varphi_1, \varphi_2]^T$ and happens along the axes of the wheels. By logical extension, the $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$ and the $\dot{\boldsymbol{\varphi}}$ indicate their time derivatives, i.e. speeds, respectively. Hence, the kinematics model may be expressed as follows:

$$\begin{bmatrix} \boldsymbol{J}_1(\beta_1, \beta_2) \\ \boldsymbol{C}_1(\beta_1, \beta_2) \end{bmatrix} \boldsymbol{R}(\vartheta_r)\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \begin{bmatrix} \boldsymbol{J}_2\dot{\boldsymbol{\varphi}} \\ 0 \end{bmatrix}, \tag{2.47}$$

whereas $\boldsymbol{J}_2$ is a diagonal matrix containing the radii of the wheels, $\boldsymbol{J}_1(\beta_1, \beta_2)$ represent the projection matrix of the rolling constraints, and $\boldsymbol{C}_1(\beta_1, \beta_2)$ include the sliding constraints. The matrices $\boldsymbol{J}_1(\beta_1, \beta_2)$ and $\boldsymbol{C}_1(\beta_1, \beta_2)$ are further defined as follows,

$$\boldsymbol{J}_1(\beta_1, \beta_2) = \begin{bmatrix} \sin(\alpha_1 + \beta_1) & -\cos(\alpha_1 + \beta_1) & -\lambda_1 \cos\beta_1 \end{bmatrix}, \tag{2.48}$$

$$\boldsymbol{C}_1(\beta_1, \beta_2) = \begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & \lambda_1 \sin\beta_1 \\ \cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & \lambda_2 \sin\beta_2 \end{bmatrix}. \tag{2.49}$$

2. **Steering Limitations:** With a steered vehicle, it is necessary to employ a control approach where the major determinants are the wheel steering angles ρ_i, for trajectory tracking or path-following task. If the steering angles remain unrestricted, such vehicles' path-following ability is minimally influenced in theory. However, the Archimede rover differs from this in that it has stringent restrictions on steering angles, as it has been described in Sec. 2.3 and the steering limits for each wheel are listed in Table 2.6. Non-holonomic motion that is spatially constrained results from this. Additionally, it can be seen that $\boldsymbol{\Omega}_r = f(\rho_i)$ if we take the rover's ICR into consideration. The fact that the ICR may be derived from $\boldsymbol{C}_1$ is evident, which in turn, through β_i is dependent on ρ_i. This can be expressed as,

$$\boldsymbol{\Omega}_r = \left[\boldsymbol{C}_1^{\dagger}\right]^{-1} \left(-\boldsymbol{C}_1^{*} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \tag{2.50}$$

where,

$$\boldsymbol{C}_1^* = \boldsymbol{C}_1 \begin{bmatrix} 0 & \dot{\vartheta} & 0 \\ -\dot{\vartheta} & 0 & 0 \\ 0 & 0 & \dot{\vartheta} \end{bmatrix} \quad \text{and} \quad \boldsymbol{C}_1^\dagger = \boldsymbol{C}_1^* \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{2.51}$$

Recalling that, as detailed described in Sec. 2.3.6, the introduction of joint limits on the steering axes of the rover has the drawback of generating surfaces of permissible ICRs location that are defined as a subset $S^+ \in \mathbb{R}^2$, and surfaces of non-permissible ICRs location defined as a subset $S^- \in \mathbb{R}^2$. In Figure 2.56, a sketchy representation is again shown in light cyan. Still, with reference to Sec. 2.3.6, it can be seen that this process generates also smaller surfaces, for example beneath the rover's belly. However, for the sake of this investigation, only the big primary triangular sections at the rover's sides are taken into account. A smooth non-holonomic vehicle motion is possible throughout operations because of the ICR's ability to travel continuously inside each of these regions.

Due to these restrictions, steering-derived motion constraints have emerged, which restrict the robot's entire mobility. In the next paragraph, it will be distinguished between the planning of the rover's orientation and the rover's path planning, which is intended to be solely positional.

3. **Path planning:** The steering, which is achieved by the determination of an appropriate and feasible ICR point $\boldsymbol{\Omega}_r$ at each time-step of the simulation, is the sole regulated feature of the path planner that has been implemented, and which is based on a constant forward-speed control. Actually, the location of the rover at $\boldsymbol{P}_r(t_0)$, which is in the trajectory $\boldsymbol{s}_r(t)$ and having a specific orientation $\vartheta(t_0)$, is shown in Figure 2.56c for the specific time step t_0. The vehicle is ordered to reach the location $\boldsymbol{P}_r(t_1) \in \boldsymbol{s}_r(t_1)$ at time step $t_1 = t_0 + \Delta t$. Any generalized planar transformation is known to be a rotation along a predetermined axis that is perpendicular to the plane. Since the vehicle must travel from $[\boldsymbol{P}_r(t_0), \vartheta(t_0)]^T$ to $[\boldsymbol{P}_r(t_1), \vartheta(t_1)]^T$, it is, therefore, possible to locate the mentioned axis on the ICR $\boldsymbol{\Omega}_r(t_0)$. In actual fact, the ICR and corresponding rotation angle ϕ may be calculated as follows:

$$\boldsymbol{\Omega}_r(t_0) = \boldsymbol{P}_0 + \left(\frac{1}{2} + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \frac{1}{2 \tan\left(\frac{\vartheta_1 - \vartheta_2}{2}\right)} \right) \overline{\boldsymbol{P}_0 \boldsymbol{P}_1}, \tag{2.52}$$

$$\phi(t_0) = \vartheta(t_1) - \vartheta(t_0), \tag{2.53}$$

where the terms $\boldsymbol{P}_r(t_0)$, $\boldsymbol{P}_r(t_1)$, $\vartheta(t_0)$ and $\vartheta(t_1)$ are denoted with the shorthand notations $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, ϑ_0 and ϑ_1. Recall now the discussion done in Sec. 2.3.6, S^+ is constrained and limited, therefore it is possible that $\boldsymbol{\Omega}_r \notin S^+$, thus incompatible with the motion. By using the ICR projection method, which has been presented in Sec. 2.3.6 a new compatible ICR point $\boldsymbol{\Omega}_r^*$ can be obtained by projecting the old one on S^+. Figure 2.56d illustrates again this specific scenario. For the sake of convenience, the cases

characterizing the ICR projection method are given again below and are in Figure 2.56c and Figure 2.56d:

Case A. $\boldsymbol{\Omega}_r \in S^+$; this means that $\boldsymbol{\Omega}_r^* = \boldsymbol{\Omega}_r$;

Case B. $\boldsymbol{\Omega}_r \notin S^+ \ \wedge \ \exists \boldsymbol{\Omega}_r$ close to $\boldsymbol{\Omega} S^+$; in this case $\boldsymbol{\Omega}_r^*$ is chosen as the closest point $\in S^+$ to $\boldsymbol{\Omega}_r$;

Case C. $\boldsymbol{\Omega}_r \notin S^+$ and $\nexists \boldsymbol{\Omega}_r$ close to $\boldsymbol{\Omega} S^+$; in this case $\boldsymbol{\Omega}_r^*$ is chosen as the tip of the S^+ area.

Unlike the other cases, case A ensures that the vehicle will arrive at point $\boldsymbol{P}_r(t_1)$ precisely. The vehicle can, however, realign itself after a certain amount of time steps Δt if $\boldsymbol{\Omega}_r^*$ is reasonably close to $\boldsymbol{\Omega}_r$.

Last but not least, the successive point of the trajectory, $\boldsymbol{s}_r(t_2)$, is chosen with a time step of $t_2 = t_1 + t_s$ whenever the rover is closer to $\boldsymbol{P}_r(t_1)$ than a tunable threshold parameter d_t. Taking into consideration these aspects, and referring to Figure 2.56 the vehicle travels along the *effective trajectory*, which deviates minimally from the *nominal trajectory* for sufficiently small Δt.

4. *Orientation planning:* Considering that in this investigation the rover is modeled as a two-wheeled steerable vehicle, it may move in any direction and in an endless number of orientations, assuming that they are consistent with the other introduced constraints. It has then taken advantage of this feature in order to provide the rover with more turning capability by enabling it to anticipate potentially impractical turns. By utilizing a moving average with a symmetric window of width $2\vartheta_{lh}$ — the subscript lh stands for *look-ahead* — it has chosen a rather straightforward strategy. Assuming $\vartheta^*(t)$ to be the orientation of the path at time t, it is possible therefore to define the planned orientation as follows,

$$\vartheta(t) = \frac{1}{2\vartheta_{lh}} \sum_{i=-\vartheta_{lh}}^{+\vartheta_{lh}} \vartheta^* \left(t + \vartheta_{lh} t_s\right). \tag{2.54}$$

5. *Control:* It has been designed, added, and tuned a proportional-derivative (PD) controller for the steering of the bicycle model. This is done in order to ensure that the simulations that will be performed will accurately reflect the behavior of the vehicle. As a result, instead of operating directly on the actual four-wheel model, the controller is operating on the model that has only two steerable wheels. It would be impossible to solve the model using a non-holonomic formulation if a four-wheel PD steering controller caused the wheels to be out of alignment. Finally, the $K_p = 2$ and $K_d = 0.05$, for both of the wheels, are the controller's gains values that have been chosen.

2.4.3 Results

In this section are reported the results that have been obtained regarding the numerical simulations of the object towed by a rover, with and without the kine-

matic constraints. Specifically, are reported first the results of the path planning and of the QSM exclusively. Subsequently, the results of the sensitivity analysis were performed leveraging the dynamics simulations. Furthermore, an in-depth discussion is presented in order to highlight the main aspects of the models' behavior.

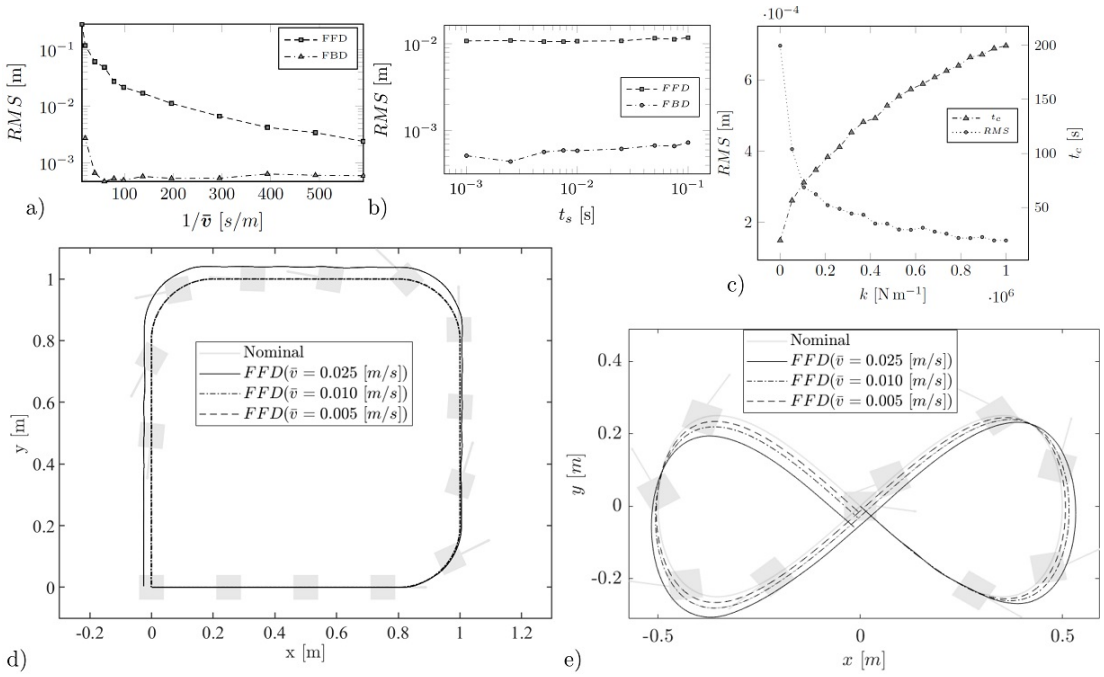


Figure 2.58: Sensitivity analysis. In a) and b) the evolution of the RMS for the two dynamics cases FFD and FBD are shown for the Lemniscate trajectory; in a) as a function of the object mean speed $\bar{\boldsymbol{v}}$; in b) against simulation time-step t_s. In c) the influence is shown of the cable spring constant κ on the RMS and on the computational time t_c for the FBD case. In d) the FFD rounded square trajectories are shown at different velocities $\bar{\boldsymbol{v}}$ compared to the nominal; in e) the same is shown for the lemniscate trajectory.

Path-planning results

For the sake of path planning, this paragraph primarily addresses the QSM findings; the dynamics on the other hand are covered in the paragraph that follows. The trajectories that have been taken into account in the present study are a diagonal line of 1.4 m long, a 1 m radius circle, a 2 m broad lemniscate, and a 4 m by 4 m square having rounded corners. Additionally, the shape of the towed body is depicted in Figure 2.54a and is selected to be a generic one. Additionally, using the body's reference frame (again as shown in Figure 2.54a), three distinct COM locations have been selected for each trajectory, these are: $G_1 = [0.05, 0.15]$m, $G_2 = [0.26, 0.15]$m and $G_3 = [0.10, 0.25]$m. This approach, namely considering a body with general geometry and unbalanced COM, has allowed to fully exploit the model's generic formulation for the continuous contact pressure distribution, shown in the previous paragraphs.

The assumptions made are that the trajectories are required to be finished in $T_{end} = 10\,\text{min}$ and a simulation time-step equals to $t_s = 1 \times 10^{-2}\text{s}$ throughout the performed simulations utilizing the QSM for trajectories synthesizing. These parameters were selected to take the QSM method into consideration.

The bodies and the towing vehicle's nominal trajectories are depicted in Figure 2.57 for each trajectory example, with the assumption that the motion is slow enough to be regarded as quasi-static. Specifically, the first row (a-d) presents the quasi-static solutions in a graphical form for the line, circle, lemniscate, and square with rounded corners trajectories with the towed object's pose. Meanwhile, the other charts (e–h) illustrate the nominal trajectories of the object being towed (black curve), as well as the towing vehicle for every of the gravity application, points G_1 (red curve), G_2 (yellow curve), and G_3 (purple curve). This has been done again for all the trajectories, namely the ones representing a line, a circle, a lemniscate, and a square with rounded corners. It is important to remark that the QSM method described in Figure 2.4.2 is used to generate these trajectories. From those findings, it is apparent that the positioning of the application point, which unbalances the object inside the bounds of the geometry of the object being towed, affects the nominal trajectory that the rover has to follow in order to manipulate the object as planned. However, it doesn't appear that the position of G alone can provide a good indicator of the rover's trajectory.

Figure 2.59 has been reported in this context in order to emphasize the impact that the introduction of the technique for the ICR projection, discussed first in Sec. 2.3.6 and then in Sec. 2.4.2, has on the trajectories. The object trajectory chosen and seen in the picture is the lemniscate curve, just as in Figure 2.57e. Moreover, both the rover's (a) and the object's (b) trajectories are depicted. It should be noted that this particular choice of path and planning parameters sets a significant emphasis on the methodology's accuracy, which in this case serves to illustrate how the planner acts throughout the process. It is therefore possible to specifically observe how it switches back and forth between cases A, B, and C multiple times, each of which is highlighted in a distinct color. As envisioned, the planner spends most of the time in case A, moving to case B or C during the most problematic portions of the path. Figure 2.59b, which shows the actual trajectory performed by the object being towed, with regard to the ideal lemniscate curve, reflects these occurrences. Therefore, it should be observed that the two charts have correspondence with the planner cases (Figure 2.59a and Figure 2.59b)

Sensitivity analysis

In this section, will be provided the findings that have been obtained from a performed sensitivity analysis. The objective is to describe how certain factors affect FFD and FBD dynamics cases. It has been chosen to concentrate the study by choosing two of the example curves, i.e. the lemniscate and the square with rounded edges ones. Additionally, it has been elected that the shape of the object being towed is a cube subjected to an eccentric load, and represents the sample geometry for the whole study. Moreover, the object's mean speed $\bar{\boldsymbol{v}}$ along the trajectory, time-step t_s of the simulation, as well as the cable's stiffness κ are the factors that have been taken into account in this investigation. Finally, the

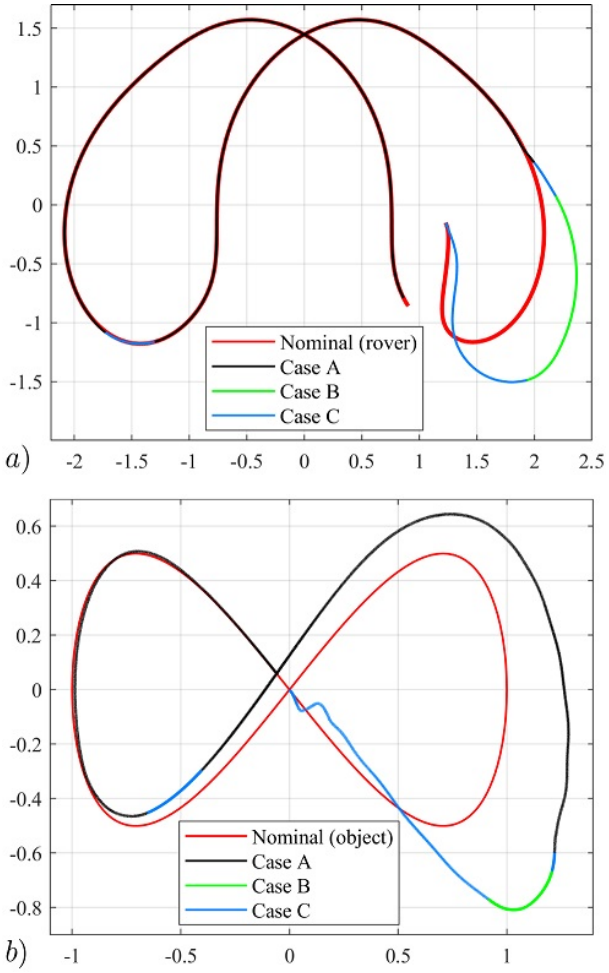


Figure 2.59: Effective trajectories of the rover and object with details on the path-planning ICR projection cases. We assumed $d_t = 0.080$. In a) the trajectories of the rover are shown and in b) we show the trajectories of the towed object. In both plots, the red line indicates the nominal path.

RMS error calculated between both the nominal trajectory generated through the QSM and the trajectories obtained through dynamic simulations has been selected as the metric value used within the presented sensitivity analysis. The dynamic simulations have been performed in the context of both the FFD and FBD cases.

Using Figure 2.58 as a guideline, it is possible to observe that a) demonstrates how, for the FFD case, the RMS diminishes significantly with decreasing the average velocities $\bar{\boldsymbol{v}}$. Nevertheless, for the case of FBD, it can be seen that the RMS remains low and nearly constant, and it does not appear to have a significant influence over the majority of the interval of velocities that have been taken under consideration. In actuality, it has been observed that this value only grows with $\bar{\boldsymbol{v}}$ values greater than $5 \times 10^{-2} \mathrm{m\,s^{-1}}$. Moreover, when the average speed is fixed to $\bar{\boldsymbol{v}} = 5.1 \times 10^{-3} \mathrm{m\,s^{-1}}$ and the time-step t_s is made varying, it is possible to observe that the value of the RMS does not vary in a considerable way. Additionally, the difference in RMS, still for the case of the time-step t_s variation

investigation, i.e. shown in (b), between the FFD and FBD cases, can be linked to the evident order-of-magnitude mismatch between the curves in a) by entering with the assumed fixed speed $\bar{\boldsymbol{v}} = 5.1 \times 10^{-3}\mathrm{m\,s^{-1}}$. Moreover, the discrepancies between both the simulated and the nominal trajectories for the towed object, using only the FFD dynamics and evaluating it for three distinct speeds $\bar{\boldsymbol{v}}$, are displayed in the graphs in Figure 2.58(d) and Figure 2.58(e), respectively for the square with rounded corners and the lemniscate curves. Specifically, both graphs show that, in comparison to the nominal curve, the dynamic curves show greater drift as the speed grows. In actuality, the trajectories converge to the nominal one whenever the speeds assume appropriate values, that are low enough, namely when the condition is converging to quasi-staticity of the motion assumption.

These results clearly demonstrate that the velocity $\bar{\boldsymbol{v}}$ does have a serious influence on the simulation's accuracy for the FFD case. This is probably caused by the inherent limitations introduced by the QSM model, which only works at arbitrarily low speeds and assumes that dynamic effects can indeed be ignored.

Additionally, the FFD simulation does have a tendency to be unstable at the beginning phase of the simulation. This is most likely due to a transient at $t = 0\,\mathrm{s}$ that is not recorded by the QSM and is associated with the object being motionless before being forced to accelerate. Due to the quasi-static nature of the approach itself that is employed, where the speed is assumed to be steady and nearly null, this variation in speed cannot be represented by the QSM. Summarizing, the FFD approach demands that the object can not be put in motion with an initial acceleration. It has therefore decided to set the object's initial speed in order to coincide with the nominal initial tracking velocity, provided by the QSM. This has been done in order to perform the dynamic simulations and has been observed that this consideration resolves the issue.

In contrast to the previous case, it appears that the velocity $\bar{\boldsymbol{v}}$ does not have an effect on the trajectories for the case of FBD dynamics. This could be because the system is described by a model which assumes that the introduced spring regulates and adjusts the object's position, due to its feedback nature. Likewise, the forces are supplied to the system in a blind way in the FFD case, which has the effect of accentuating deviations. Taking into account the just mentioned considerations, Figure 2.58c illustrates how the spring stiffness κ affects the RMS metric as well as the computational time t_c in the case of the FBD dynamics simulations. As predicted, this parameter seems to have a significant impact on both observed parameters, with greater κ values producing less RMS but longer calculation time t_c. This finding suggests that the model does have some numerical stiffness.

When the rover's mobility is determined by the kinematics that has been outlined in Sec. 2.4.2 rather than being omnidirectional, a similar strategy is used. Indeed, both positional path and orientation plannings are essential for successful path-following task, according to the hypotheses that have been presented in Sec. 2.4.2. How the parameter d_t influences the trajectories of the object being towed is depicted in Figure 2.60. Specifically, it is evident for the majority of cases from the second set of graphs how values within the range of 0.1 m and 0.4 m produce the best outcomes. Interestingly, the lemniscate curve example exhibits

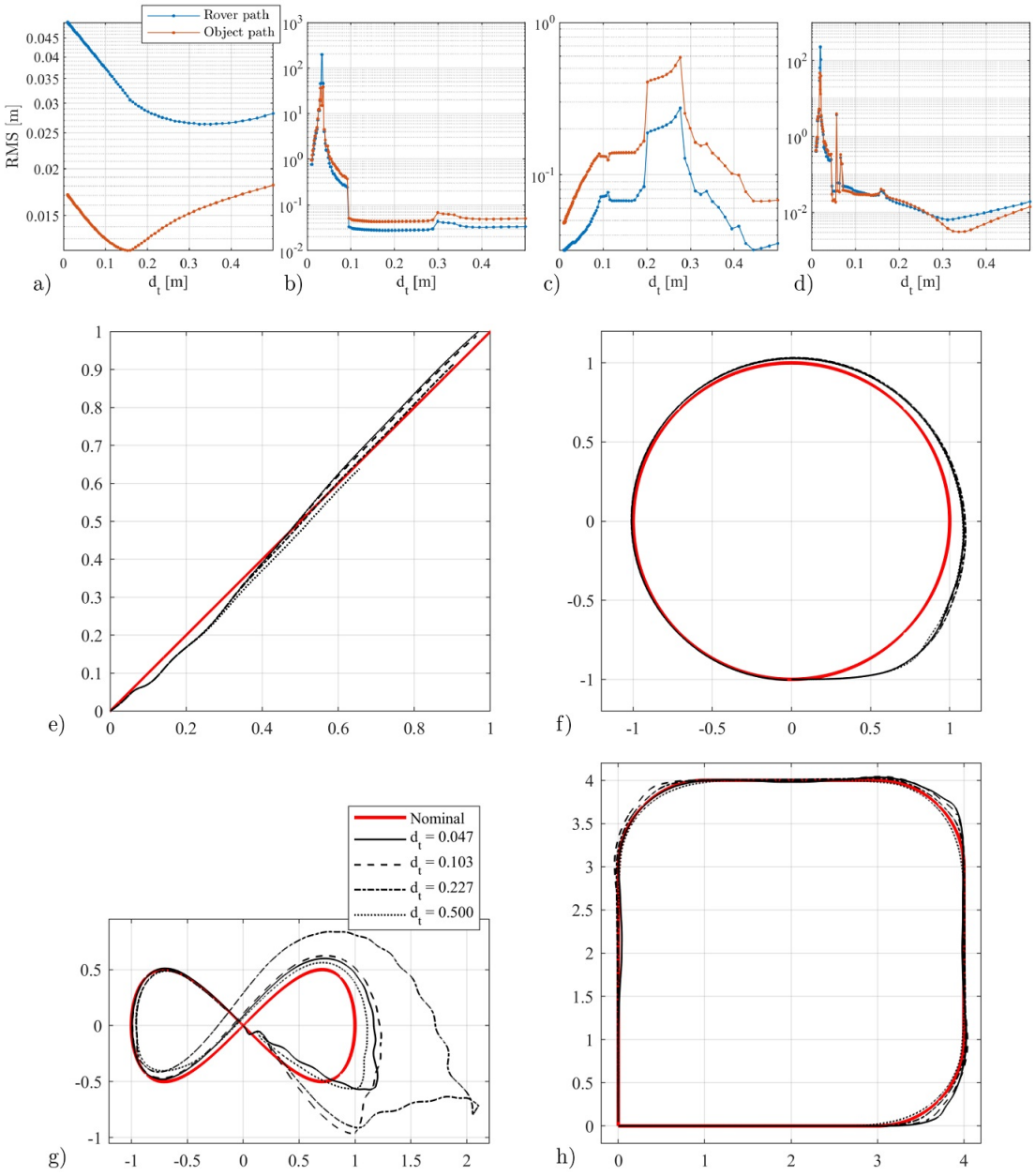


Figure 2.60: Towed object trajectories after application of kinematics constraints on the rover. The relation between the RMS value between the nominal curve and the simulated ones for the four curves: a) line, b) circle, c) lemniscate, and d) rounded square. In each plot, the RMS values are shown for the rover and towed object trajectories. Select values of d_t and the related paths are shown in e) to g) for the line, circle, lemniscate, and rounded square trajectories.

more erratic behavior. This is presumably so because the introduced controller can not operate effectively with such tiny curvatures. It can be seen from the same figure that the trajectories regarding the values $d_t = 0.103$m (dashed curve) and $d_t = 0.227$m (dash-dotted curve) totally "break" the proposed controller; the significant deviations are brought about by the fact that the calculated ICRs are quite far from the ideal one and lead the rover to deviate from more rational

paths by a very big margin.

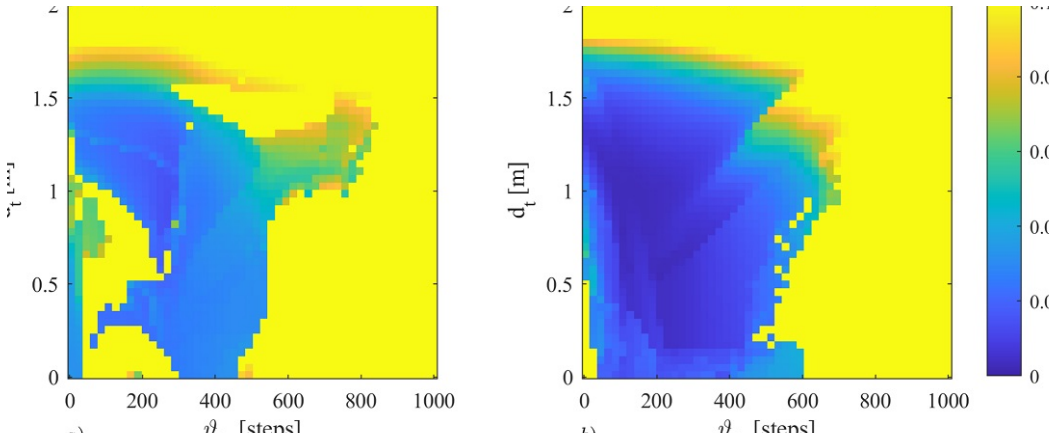


Figure 2.61: Response surface for the RMS by varying the threshold distance d_t and the orientation look-ahead ϑ_{lh} for the lemniscate (a) and rounded square (b) paths.

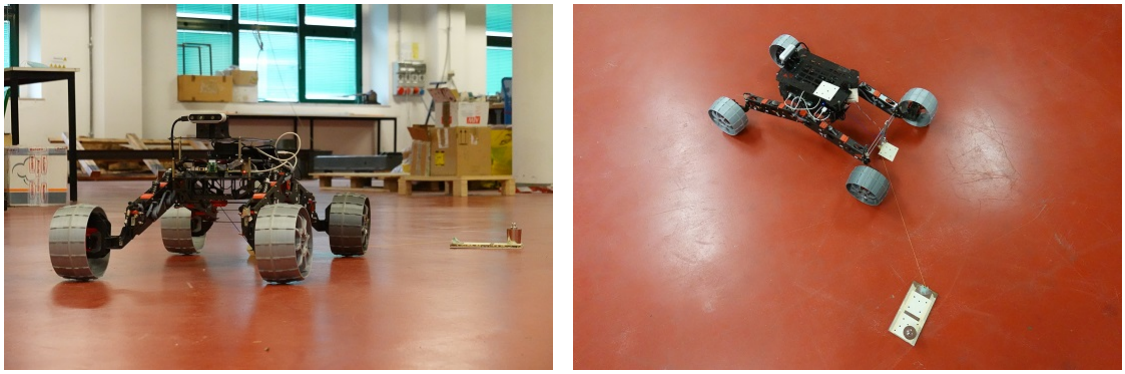
It is possible to observe how these path-planning variables affect the RMS metric for the lemniscate and rounded square trajectories from Figure 2.61. As could be predicted, the former's viable area is smaller, more ragged, and overall has higher RMS values than the second. In general, d_t ranging from 0 m to 1.5 m and ϑ_{lh} values in the range of 300 and 400 steps seem to yield feasible results.

2.4.4 Experimental validation

Within this study, the experimental Archimede rover prototype has been used to tow an object along random trajectories commanded by the user through the ROS network. The setup of the experimental test bench and the further validation of the proposed approach are provided throughout this paragraph with an emphasis on the outcomes. Specifically, the dynamics model presented in Sec. 2.4.2 has to be validated. Through direct experimentation, the coefficient of friction acting between the floor and the object contact patch has been estimated to be nearly $\mu = 0.34 \sim 0.36$. In particular, Figure 2.62 illustrates the experimental setup including two images showing the rover dragging the payload. Moreover, a little steel weight is posed above the back corner of the sample rectangular body in order to obtain the desired eccentric force. The ROS ecosystem is used to control the rover, through appropriate ROS packages and a MATLAB Desktop application, as shown in Sec. 2.3.

A Sony RX100 VII camera mounted on the ceiling is used for data acquisition. After the video feed has been calibrated, the tracking process is carried out using MATLAB's implementation of the Kanade-Lucas-Tomasi (KLT) technique [289].

Following minor pre-processing, needed to remove tightly packed point clusters, the tracked trajectory of the experimental rover has been used to serve as input for the FBD dynamics simulator without any additional constraints, for comparison and validation purposes. The object's reconstructed trajectory obtained from the experimental test is successively compared to the one obtained



(a) (b)

Figure 2.62: Experimental test of the rover Archimede towing an unbalanced object from a front three-quarter view (a) and a top view (b).

via FBD dynamics and expressed in terms of the body's geometric center trajectory. Additionally, Figure 2.63 illustrates the comparison performed between the two curves and reveals a striking conformance between both the simulated outcome and the experimental observations. The relatively minor discrepancies between the two curves are most likely caused by ground imperfections as well as the elasticity of the simulated tether.

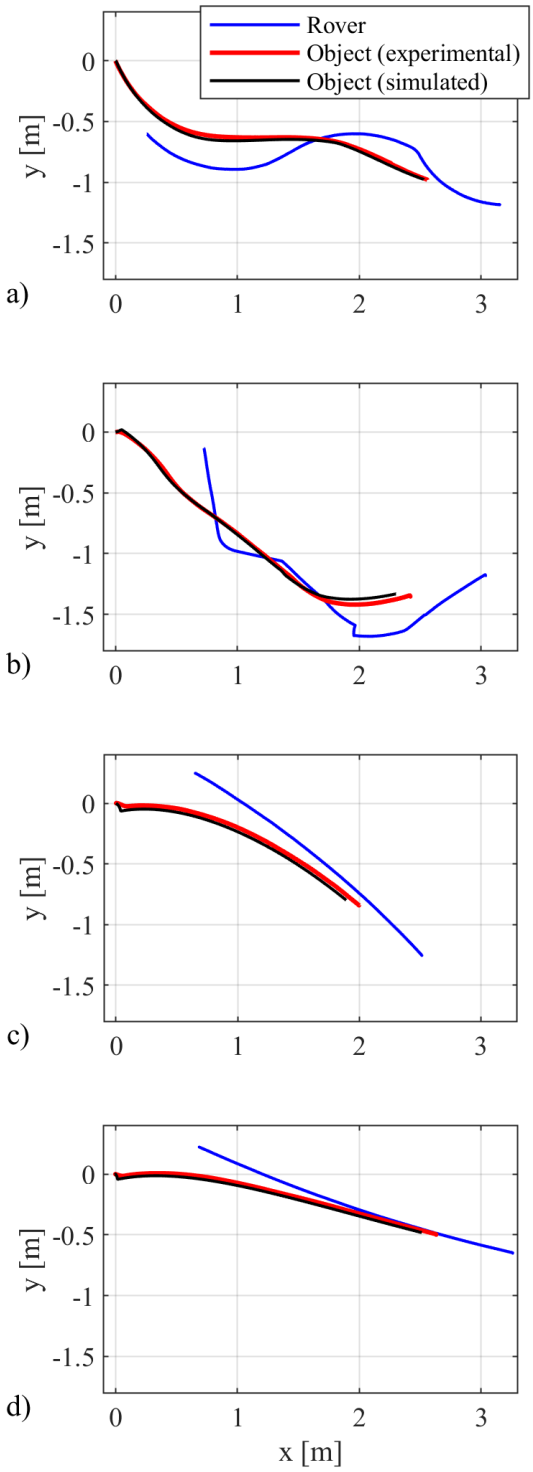


Figure 2.63: Experimental comparison between simulation and experimental results. The letters a) through d) show four diverse rover paths (in blue) that lead to simulated (black) and experimental paths of the dragged object (red).

Part II

Mobile Robotics for Industrial Applications

Chapter 3

Mobile Robotics for Industrial Applications

3.1 Introduction

With each passing day, mobile robots are being used more and more in industrial settings, as well as in other realities such as hospitals, and performing different operations. This is a consequence first of the ever-increasing of available technologies and automation, and secondly of the manufacturing industries transitioning to the Smart Factory status. Due to the requirements imposed by the Smart Factory concept, which have been outlined in Sec. 1.3.2, specifically due to the flexibility requirement and due to the problem of the plant in constant evolution, it follows the increasing need of the use of AMRs which are perfectly suited in these settings. The aim of these robotic systems is to perform value-added operations and to remove human operators from repetitive or dangerous operations.

In these settings, the mobile robots are allowed to move in the entire production plant, while trying to accomplish the assigned task. Due to their flexibility coming from their high mobility capabilities, namely capable of working in unlimited operational space, these robots may encounter unforeseen situations around the plant. It follows that these systems must be reliable and flexible, thus equipped with controllers and strategies which can handle those situations. Due to these considerations, much effort and research have been conducted on the development of fully autonomous systems, strategies, and more and more sophisticated controllers.

Moreover, these robotic systems in the Smart Factory settings are currently used for loading/unloading operations, material transporting and intralogistics operations in general. These mobile robots are managed and coordinated in these settings by a fleet manager. However, the current iteration is to assign more responsibilities to these robotic systems, hence allowing them to perform more complex operations and tasks. It follows that, depending on the nature of the task assigned to them, they might need to perform it in a cooperative way with the other mobile robots present in the production plant. Thus a safe and smart multi-robot approach is needed.

Moreover, in Smart Factories, mobile robots are required to operate in an

environment that may or may not be populated by human operators. These mobile robots may be asked to perform a task, but at the same time be aware of the presence of human operators. Or they may be asked to perform operations collaboratively with human operators. It follows that these systems must possess advanced controllers and provide a high level of autonomy to ensure and deal with these situations. Consequently, they must be able to guarantee high levels of safety and therefore be unable to cause harm to humans.

Taking the above considerations into account, it follows that these mobile robots are subject to all three types of interaction that have been described in section Sec. 1.2. Specifically, mobile robots are subject to interactions with their surroundings, for example, contact with the ground beneath it, proximity to the obstacles, or with the other network-connected machinery and infrastructure; to robot-robot interactions in the context of the mobile robot fleet and in collaborative multi-robot applications, or to human-robot interactions that are either just in being aware of their presence and avoiding them, or while performing tasks collaboratively. However, differently from the case of the space application mobile robotics, outlined in Chapter 2, the environment-robot interaction is not the most critical one anymore and which must be taken into account first. As an example, in this case, usually, the contact interaction between the wheels and the floor is not so relevant since no complex phenomena can occur. The only relevant environmental interactions are the ones resulting from the obstacles placed inside the plant. Differently, the robot-robot interactions and the human-robot interactions for these settings are the most critical and need to be addressed correctly.

Considering what has been described and mentioned above, this chapter aims at addressing some of the described issues and challenges that characterize the introduction of mobile robots in industrial settings, specifically in Smart Factories. Specifically, in Sec. 3.2 will be presented a fleet manager specifically for a fleet of tethered mobile robots. The fleet manager leverages a kinematic analogy between a chain of tethered mobile robots and a redundant manipulator in order to manage and coordinate the motion of the fleet. This section will first provide an introduction and an in-depth state-of-the-art analysis specific to the problem. Subsequently, it will be described the kinematic analogy between a chain of tethered robots and a redundant manipulator and the framework to exploit this analogy will be formally modeled. Subsequently, an industrial request case study will be taken into account and a description of the constraints, additional tasks, and control architecture will be provided. Finally, the performances of the fleet manager, i.e. the controller, will be validated numerically for four different scenarios and subsequently validated in the Gazebo dynamic simulator.

In Sec. 3.3 instead will be presented the development of a controller for an AMR, while addressing the problem of human-robot interaction. Specifically, the developed controller aims at providing the mobile robot with the capability to navigate in moving crowds in a safe way. This section will first present an introduction and an in-depth state-of-the-art analysis specific to this problem. Subsequently, the modeling of the problem, which comprises the modeling of the moving crowd, the robot, and the perception system, in the simulated environment is presented. Following, it is presented the reinforcement learning

architecture, the chosen algorithms, and the elected convolutional neural network topology will be described. Moreover, will be presented the training and validation results obtained for all the elected reinforcement learning algorithms, hence numerically validate the approach. Finally, will be presented the experimental validation of the approach.

3.2 Tethered mobile robots

In this section, we aim to study and address an application of great interest in the field of mobile robotics operating in the industrial environment. In particular, as will be seen below, the application that will be addressed in this section falls within the macro field of cooperation between multiple mobile robots.

A key idea in the field of industrial mobile robotics is to give mobile robots the ability to perform an increasing number of tasks, even the most critical and resource-intensive ones, thus expanding the range of their possible applications. In addition, if we consider a Smart Factory, as was mentioned in Sec. 1.3.2, these are highly robotic, highly flexible environments with constantly evolving facilities. With reference to robotic applications employing mobile robots, multiple mobile robots, controlled by the so-called fleet manager, are usually employed in these particular environments.

Taking into account the considerations just stated, this investigation aims to present the study of an approach to synthesizing a controller with route planning capabilities, i.e., a fleet manager, for a fleet of mobile robots arranged in an open chain in the environment. There are applications that require mobile robots to be somehow connected to each other through a connector element. As will be seen in this section, particular applications, which require these connection requirements, are those that require the redistribution of electricity and power, or fluid material between an origin point and the mobile robots.

Specifically, a formal general methodology will first be presented that allows one to kinematically consider a chain of mobile robots, mutually connected with flexible elements such as cables, as an equivalent redundant robot manipulator, and thus exploit their redundancy for the control of the entire chain of robots. Subsequently, the focus of this study will be on the application and validation of this methodology in concrete application examples in the industrial field. In particular, one of the studies performed in order to validate the model takes as an application example the robotic painting of large objects, such as boats and small airplanes, which do not have precise positioning in the work environment. This study shows how the proposed methodology makes it possible to develop and generate a fleet manager for hypothetical application contexts, in Smart Factories, of chained mobile robots.

Specifically, in the problem of interconnected mobile robots, the interactions that occur on the generic mobile robot are multiple. Among the group of environment-robot interactions, we find the usual interaction between the robot and the ground, and the interaction that occurs with the surrounding environment, such as obstacles present. Moreover, in this study, the robot-robot interaction is also verified as the mobile robots are connected to each other. Therefore

the generic mobile robot interacts with the other robots in the fleet through the cables that connect them. It is useful to report that for the purpose of this study and thus to the development of a methodology and fleet manager, the interaction that the mobile robot undergoes through contact with the ground is irrelevant, so it will not be considered. Furthermore, the interactions that the robot has with objects in the work environment and that it has with other robots will be encoded directly as additional tasks assigned to the high-level controller, i.e., the fleet manager. Hence, in this section, it will also be seen how these interactions are taken into account and subsequently how they are encoded as additional tasks.

Summarizing, the contributions of this secton can be listed in:

- The formal formulation of a kinematic analogy between a chain of tethered mobile robots and a redundant manipulator
- The use of the developed framework to address a particular industrial problem, thereby assessing the viability of a chain of mobile robots and finally the validation of the framework;
- The development of ROS packages for dynamic simulations in ROS-Gazebo and validation of the approach.

3.2.1 Introduction

One of the most important characteristics of mobile robotic systems since their inception has been, above all, their autonomy, both decision-making and physical. This means that, in a broad sense, a robot with mobility should be physically unconnected to its surroundings or other fixed equipment. Modern energy storage technologies, such as batteries, remote control, or autonomous onboard decision-making, have been created and put into use as a direct result of this requirement [290].

Having said that, there are some particular situations in which a physical link to the mobile robot is required or at the very least very desired. For instance, these could be:

- applications requiring a lot of power and energy consumption, such as, for instance, those for welding or perpetual drone flight;
- exploration of areas that prohibit telecontrol, such as underwater environments, mines, tunnels, and so on;
- transmission of materials, including for instance pressurized air and paint for robotized painting applications, from a base station towards the robot, or payload transportation with drones;

Regardless of the fact that literature on single Tethered Mobile Robots (TMRs) is extensive, this study has posed attention to multi-robot systems such as the platoons [291] and even trains of mobile robots which are connected by cables [292–294]. The first group deals with TMR arrangements in a parallel manner,

whereas the second deals with mobile robot chains or made it simple, in serial arrangements.

It is apparent that the amount of workspace and maneuverability may be greatly increased simply by connecting many mobile robots such that they form a chain. Previous studies on this type of system tended to focus mostly on control-related issues and frequently lacked a clear kinematic description. By considering a network of TMRs, arranged such that they form a chain, as a serial robotic manipulator, here is provided a formal method to manage the kinematics of these types of systems. Thus, the cable is viewed as a prismatic joint and each mobile robot is represented as a revolute or spherical joint. From now on and for the whole study, the leading robot is also referred to as the *end-effector*, the robots in the chain in between are referred to as the *intermediate robots*; and finally, the fixed connection point of the chain is referred to as the *base*.

Formally, the equivalent robotic manipulator turns into a redundant system whenever there are two or more mobile robots. This similarity enables to use of the redundant manipulators (RMs) inverse kinematics (IK) methods to address the problem. In this context, limitations on the kinematics of mobile robots are stated as supplementary tasks the equivalent RM has to take care of. One of these constraints, for instance, is indeed the tether's influence because, for example, it must not come in contact with the ground.

This investigation has been inspired by an unique industry need for painting robot applications on very large objects, such as small boats or aircraft. In this instance, the robotic painting arm needs to be able to maneuver in a space in which the objects are also not precisely positioned. Furthermore, robotized painting faces two primary issues: first, it necessitates a significant amount of power, therefore it is plainly impossible to complete the assigned task by using just batteries; second, the working atmosphere will be hostile which translates into the fact that operators can not be physically there. Pressure washing, on the other hand, is also a significant application of TMRs arranged in chains within this context. However, a direct connection from the wall up to the employed mobile manipulator, for this task, is very unfavorable because the manipulator has the need to traverse a large area to accomplish the assigned task. Indeed, the cable would run into the obstruction or tangle in this scenario because the mobile manipulator has to be able to maneuver around the obstruction. This is why utilizing several mobile robots arranged in a chained series is a solution, in addition to improving the system's maneuverability. The purpose of the intermediate robots is solely to change the workspace's cable layout and behave as a *fairlead*. However, the addition of cables to the environment significantly complicates planning, control, as well as their management throughout the system's entire motion.

It is acknowledged that the literature on chains of TMRs often lacks precise kinematic formulations; more precisely, the majority of contributions found are restricted to the field of the control theory facets. The purpose of the current research is mainly to address the industrial challenge while at the same time offering a broad framework wherein TMRs are exhaustively and formally elegantly characterized as equivalent redundant manipulators.

The framework that will be later described makes it exceptionally simple to integrate the cable since it enables a description of the cables based on the catenary formulations, eliminating the notion of restricting taut cables while also allowing for the computation of tension forces as well as the implementation of kinematic limitations on the robots. Specifically, by associating the tether to a link of a robotic manipulator, it is therefore feasible to constrain the tether such that it never touches the ground and so that its elongation is always contained within a specific range by using the proper techniques. Furthermore, the standard manipulator's obstacle avoidance technique makes it possible to reconfigure the cable in the surroundings in a practical manner. With this strategy, the planning problem of the mobile robots is made easier because the tether may avoid touching the ground and won't tangle in the surroundings. Finally, the framework makes it simple to iteratively extend the system of tethered robots to the scenario of n robots in a general way. It is believed that this framework will find many application scenarios and will provide many benefits due to the introduced simplification of these extremely complex systems.

Within this study, a sensitivity analysis has been conducted on the presented framework's key parameters in order to properly define it, revealing the methodology's benefits and shortcomings. Furthermore, it has been elected to use the Gazebo dynamics simulator in order to simulate robot movements and validate the outcomes. This has provided total control over the environment and successively helped rule out exogenous error causes associated with unstructured, real-world situations.

Finally, this section is structured as follows: in Sec. 3.2.2 is provided an in-depth state-of-the-art analysis on the broad topics of the tethered mobile robots and the redundant manipulators. In Sec. 3.2.3 is described in detail the main methodology used within this investigation, i.e. the kinematic analogy between a chain of TMRs and a redundant manipulator, hence the framework will be provided. In Sec. 3.2.4 is described the case study that has been taken into consideration, in which the chain of TMRs is composed of three omnidirectional mobile robots. In Sec. 3.2.5 are reported the description of the primary task and the additional tasks that will be included in the presented framework, in order to accomplish the given mission requirements. In Sec. 3.2.6 is described the control architecture used, and that defines the fleet manager. Finally, in Sec. 3.2.7 are reported the results of the application of the methodology for the case study considered in four different scenarios. Subsequently, are provided the results of a sensitivity analysis that has been performed in order to evaluate the influence some hyper-parameters of the model have on the controller performances. Moreover, the correctness and soundness of the presented methodology will be evaluated by performing multi-robot simulations in the Gazebo environment.

3.2.2 Related works

The two key state-of-the-art components that are pertinent to this study, which will be presented in the next sections, are addressed below. These can be classified into mobile robots which are linked together (tethered robots) and strategies and

methodologies used for solving robot redundancies.

Tethered robots

Tethers are a desirable feature in situations in which mobile robots have significant power requirements; these elements are also frequently used for data, communication, and control in those situations where exceptionally high levels of reliability are required. Underwater robotics [295] ROVs (Remotely Operated Vehicles) are a prime demonstration of this, as wireless communications with the mother ship through water are frequently impractical, thus a physically linked one is needed. The literature demonstrates applications in matter transfer, as an instance for fluids [290], including towing materials on the ground [296] and in the air, [297–299], for robotized tank inspection [300] as well as traction support and assistance for mobile robots ascending steep slopes [301–304]. They are also utilized in mobile CDPR [305], found significant applications in the field of Urban Search and Rescue (USAR) operations [290] and in Unmanned Aerial Vehicles (UAVs) [306].

In 2001, Fukushima et al. examined TMRs that used a cable model known as a "hyper-tether". The length and tension of the tether may be actively controlled using this methodology [307]. Later, in 2014, Kim et al. addressed the problem of a TMR's navigation and created a topological technique that is based on the cable's homotopy class [308]. More recently, a multi-heuristic A^* algorithm method has been used by Kim and Likhachev in 2015 to investigate the problem for the path planning of a TMR and shown in [309]. Additionally, TMRs have been used in space missions, such as the planetary rovers Nanokhod [290] as well as the ESA ROSA MicroRover [310], where the mission requirements for the rovers' size and mass make a fully autonomous system unfeasible.

A significant amount of research contributions can be found in the literature regarding a single mobile robot that is tethered to a fixed location, on the other hand, exist just a few examples of chains of TMR. As an instance, Tognon and Franchi addressed two tethered multicopters linked with a base and considered the overall system as a 2DOF planar arm [293]; furthermore, on the same line, Kosarnovsky and Arogeti take into account an arbitrary number of UAVs as a robotic arm [294]. Moreover, other works suggest a model to control a sequence of UAV multicopters that are linked with one another by flexible tethers [292].

Approaches towards robot Redundancy

It is very beneficial that the IK for a RM offers infinite solutions since redundancy may be used to carry out additional and multiple tasks or to impose on the system extra constraints. The *Extended Jacobian* (EJ) technique [311], the *Task Space Augmentation* (TSA) [312, 313], the *Task Priority* (TP) approach and the *Gradient Projection Method* (GPM) are the primary approaches that had been developed to address the redundant kinematics while fulfilling these additional tasks and constraints. Specifically, both TSA and EJ seek to expand the Jacobian by including the equations for the constraints. The constraints are represented differently in each of these ways, and the IK solution is achieved simply by invert-

ing the new Jacobian. It must be pointed out that algorithmic singularities can occur with any approach. More in detail, the risk that a singularity condition occurs, grows as the number of additional tasks increases. For instance, even though the Jacobian of the single tasks has full rank, the augmented Jacobian can become singular [314]. Furthermore, Simas et al. suggested a technique to modify the coefficients characterizing the EJ's constraints in order to prevent the robot from experiencing singularity conditions [315].

The TP method had been first put out by Nakamura et al. [316] and Maciejewski et al. [317] for the case of two tasks, and it was then extended generally by Sciavicco et al. to that of the case of t tasks [318]. Despite this, the approach is vulnerable to algorithmic singularity [314], which may be avoided by utilizing Chiaverini's similar formulation [319, 320] that is more stable near singularity conditions of the Jacobian but comes with the drawback that exhibits a higher tracking error for auxiliary tasks. Furthermore, this approach had been further extended in order to deal with t multiple tasks by Baerlocher et al. [321]. Last but not least, GPM [322, 323] is a technique that maximizes an objective function in computing the joint variables. The internal motion of the robotic RM is successively determined by the projection of the vector into the null space of the end-effector's Jacobian.

Platooning issues and formation control have been addressed using methods established for conventional RMs [324–326]. In other research by Antonelli et al. [327, 328], task conflicts were avoided by employing, built on previous works, singularity robust task priority [319, 320].

The Jacobians in the task space are in general computed in earlier studies with regard to the robot's location. In contrast, in this study a virtual RM is explicitly controlled; as a result, the Jacobians are calculated with regard to the virtual RM's joint variables. In other words, although the locations of the single mobile robots are mapped by using the direct kinematics of the RM, the virtual RM is controlled directly in the joint space.

3.2.3 Methodology

In this paragraph, the general methodology used to solve the problem of controlling a chain of n TMRs is presented.

All navigation problems, in principle, can be decoupled into three different layers of abstraction: path planning, kinematics model, and actuation. Here the attention will be focused on the first. Indeed, it is assumed that the robots have their own separate controller which takes care of tracking an assigned reference trajectory.

Consider a chain formed by a number n of mobile robots which are distributed in three dimensions, with the initial one being connected to a fixed location (such as the ground or a wall) and the others being interconnected in order to constitute an open chain. Then, the vector $\boldsymbol{x_k} \in \mathbb{R}^3$ and $k = 1, \ldots, n$ determines the location in the space of each of the k-th mobile robot. The fact that now the mobile robots are linked together in this chained manner causes the k-th robot to be dependent on the robot before it, namely, $\boldsymbol{x_k} = f_k(\boldsymbol{x_{k-1}})$.

As illustrated in Figure 3.1, the mentioned analogy entails treating every mobile robot by modeling as a spherical joint and the tether linking two successive mobile robots as either a prismatic joint. The tether form may also be addressed as a complementary problem. Since it is assumed that the cable is not spinning, the spherical joints are assumed then to have only 2 DOFs. As a result, the open chain of TMRs may be seen from a kinematic point of view as an open-chain robotic manipulator, thus making it easier to tackle the n TMRs challenge using manipulators' already established and used approaches.

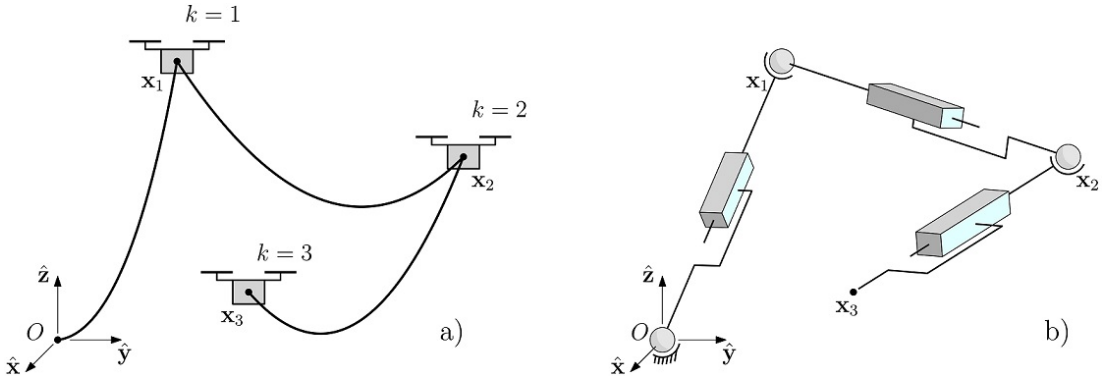


Figure 3.1: Kinematic analogy between a manipulator and a chain of TMRs: in (**a**) A view in the 3D space of a system of tethered drones forming a chain; in (**b**) its modeling as a robotic arm [329].

In order to make use of the general framework for manipulators and especially for a convenient and easy definition of the Jacobians, the Denavit-Hartenberg (D-H) convention is also used to represent the equivalent manipulator joint variables. As a result, the transformation matrix $\boldsymbol{T_i^{i-1}}$ between the two joints i and $i-1$ may be written as follows,

$$\boldsymbol{T_i^{i-1}} = \begin{bmatrix} c\vartheta_i & -s\vartheta_i c\alpha_i & s\vartheta_i s\alpha_i & a_i c\vartheta_i \\ s\vartheta_i & c\vartheta_i c\alpha_i & -c\vartheta_i s\alpha_i & a_i s\vartheta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where d_i are the prismatic and ϑ_i the revolute joint variables, and s and c, respectively, designate the sine and cosine trigonometric functions.

It is important to remark that a 2 DOFs spherical joint will be considered since it is assumed there is no rotation around the tether. This kind of joint can be easily considered as two planar revolute joints with orthogonal axes intersecting at one common point. Using the D-H convention this is equivalent to having a system consisting of one link with zero length and two planar revolute joints. This leads to the consideration that there are therefore three active joints between each mobile robot. Therefore, by designating as $\boldsymbol{T_{i,j}^{i-1,j}}$ the homogeneous transformation in between two generic successive joints i and $i-1$ along the j-th connecting link which links the two mobile robots, then the pose of the k-th robot with respect to the base reference frame can be described as follows,

$$\boldsymbol{x_k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left(\prod_{j=1}^{k} \prod_{i=1}^{3} T_{i,j}^{i-1,j} \right) \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{3.2}$$

Eq. (3.2), which may be used to map the end-effector pose inside the task space, given the joints vector $\boldsymbol{q} \in \mathbb{R}^n$ which is made up of the joint variables, defines the function of the direct kinematics for the whole robotic manipulator when $k = n$. In this situation, this function has the role to enable the mapping of the configuration of all mobile robots. Furthermore, in a more condensed form, the function for the direct kinematics may be written as,

$$\boldsymbol{x} = \boldsymbol{k}(\boldsymbol{q}) \tag{3.3}$$

where $k(\cdot)$ is a non-linear operator that is defined as $k : \mathbb{R}^n \to \mathbb{R}^m$ and $\boldsymbol{q}$, as before is the joints variable vector $\boldsymbol{q} = [q_1, \ldots, q_n]^T$.

Specifically, the system's number of DOFs is increased by a factor of three for each new TMR that is added to the chain: indeed, the spherical joint accounts for 2 DOFs while the prismatic one accounts for one DOF. Moreover, to fully define a specific task in the three-dimensional task space, then in general six DOFs are required: these are the position (x, y, z) as well as the orientation that can be defined for example with the Euler angles $(\alpha, \vartheta, \varphi)$ with respect to a fixed frame. Subsequently, if the task requires only placement, it is apparent that the number of DOFs needed to adequately specify it can be decreased to three, which means that the three orientations can assume any value. Therefore, it is possible to make this assumption because it is believed that the end-effector of the equivalent robotic manipulator, i.e. the final robot of the chain, may freely spin around its axes.

The case where $n = 1$ consists of a system with exactly three DOFs, i.e. the configuration of the system has an unique solution, given the end-effector pose. This case is trivial, as such it will not be considered in this study. Thinking instead about the scenario in which $n > 1$: in this instance, the overall system is said to be redundant. Thus, $r = 3(n - 1)$ may be used to quantify the quantity DOFs which are redundant, called also *Degrees of Redundancy* (DORs). This means that the redundant system offers ∞^r solutions for a given configuration. A large number of DORs r allows the addition of a certain amount of additional tasks or constraints that will affect the system's behavior in a different way.

Consider now the relationship between the task space and the joint space velocities, known as differential kinematics, which can be obtained by differentiating Eq. (3.3) with respect to the time variable; and it can be expressed as follows,

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.4}$$

where $m = 3$ in this instance, $\dot{\boldsymbol{x}} \in \mathbb{R}^m$ represents the vector of task space velocities, $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ represents the time derivative of the vector of the joints variables, and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ represents the Jacobian matrix. It is important to point out that for redundant manipulators, the Jacobian $\boldsymbol{J}(\boldsymbol{q})$ is not squared, thus

not directly invertible, since $m < n$. In this case, the generic inverse differential kinematics solution is stated as follows for a given task space velocity vector $\dot{\boldsymbol{x}}$:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}} + (\boldsymbol{I_n} + \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))\dot{\boldsymbol{q}}_{\boldsymbol{0}} \tag{3.5}$$

where,

$$\boldsymbol{J}^{\dagger}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^T(\boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{J}(\boldsymbol{q}))^{-1} \tag{3.6}$$

$\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ is usually called as the Moore-Penrose left pseudoinverse. Since the aforementioned pseudoinverse is numerically unstable close to the singularity configurations, within this study it won't be used that formulation for future needs. On the other hand, it will be used, instead, the *Damped Least Square Method* in order to compute another pseudoinverse $\boldsymbol{J}^{\#}(\boldsymbol{q})$ and is defined as follows,

$$\boldsymbol{J}^{\#}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^T(\boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{J}(\boldsymbol{q}) + \lambda\boldsymbol{I_n}) \tag{3.7}$$

where $\lambda \in]0, 1[$ is a tunable constant which helps avoiding algorithmic singularities. The aforementioned pseudoinverse has the effect to represent a trade-off between a smoother output and stability in the close proximity of the singularities. Indeed, the addition of a constant and always positive diagonal matrix has the effect that $\boldsymbol{J}^{\#}(\boldsymbol{q}) \approx \boldsymbol{J}^{\dagger}(\boldsymbol{q})$, and will be always invertible.

Eq. (3.5)'s first term is usually referred to the particular solution that resolves Eq. (3.4) in terms of least squares, whereas its second term is also referred to the homogeneous solution. The orthogonal projection operator $(\boldsymbol{I_n} + \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))$ is an operator which will project a general vector $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$ into the null space $\mathcal{N}(\boldsymbol{J})$ of $\boldsymbol{J}$. In fact, by multiplying both terms of Eq. (3.5), it can be demonstrated that the second term produces no end-effector motion, rather, it generates only internal motions which reconfigure the system while keeping the end-effector stand still.

Within this study, it will be used Chiaverini's formulation [319, 320], or more specifically its t tasks generalization [321], in order to address the problem of the IK for a RM. Lower values denote higher priority inside this formulation's priorities, which are specified by $i = 1, \ldots, t$ and where $\sum_i^h m_i \leq n$. It is therefore possible to define the joint space velocity vector, by using this formulation, as follows:

$$\dot{\boldsymbol{q}} = \sum_{i=1}^{t} \left(\boldsymbol{P}_{\boldsymbol{i-1}}^{\boldsymbol{A}}(\boldsymbol{q})\boldsymbol{J}_{\boldsymbol{i}}^{\#}(\boldsymbol{q})\dot{\boldsymbol{x}}_{\boldsymbol{i}} \right) + \boldsymbol{P}_{\boldsymbol{t}}^{\boldsymbol{A}}(\boldsymbol{q})\dot{\boldsymbol{q}}_{\boldsymbol{opt}} \tag{3.8}$$

where,

$$\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{A}}(\boldsymbol{q}) = (\boldsymbol{I_n} - \boldsymbol{J}_{\boldsymbol{i}}^{\boldsymbol{A}^{\dagger}}(\boldsymbol{q})\boldsymbol{J}_{\boldsymbol{i}}^{\boldsymbol{A}}(\boldsymbol{q})) \tag{3.9}$$

and,

$$\boldsymbol{J}_{\boldsymbol{i}}^{\boldsymbol{A}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J_1}(\boldsymbol{q}) \\ \vdots \\ \boldsymbol{J_i}(\boldsymbol{q}) \end{bmatrix} \tag{3.10}$$

where $\boldsymbol{P}_{\boldsymbol{i}}^{\boldsymbol{A}}(\boldsymbol{q})$ is the projection operator into the $\mathcal{N}(\boldsymbol{J}_{\boldsymbol{i}}^{\boldsymbol{A}})$, i.e. the "Augmented Jacobian" that is augmented by the Jacobian $\boldsymbol{J_i}(\boldsymbol{q})$ associated to the i-th additional task.

Consider the fact that when $t = 1$, that is when the task assigned is referred to the primary one, the projection operator $\boldsymbol{P_0^A}(\boldsymbol{q}) = \boldsymbol{I_n}$. Let $\boldsymbol{J_i}(\boldsymbol{q})$ be the Jacobian related to the i-th task, and let $\dot{\boldsymbol{x}}_{\boldsymbol{i}}$ the i-th task speed vector. It should be noted that $\dot{\boldsymbol{q}}_{\boldsymbol{opt}}$, i.e. the joint optimization term, has been explicitly specified in Eq. (3.8). Now, a vector $\boldsymbol{q_{opt}}$ can be defined in such a way that it complies with a specific task or constraint. A common option after the GPM is to provide the vector in such a way that it depends on the gradient of a suitable objective function [322], that is from now on referred to as $w(\boldsymbol{q})$, therefore:

$$\dot{\boldsymbol{q}}_{\boldsymbol{opt}} = k_0 \left(\frac{\partial w(\boldsymbol{q})}{\partial \boldsymbol{q}} \right)^T = k_0 \nabla w(\boldsymbol{q}) \tag{3.11}$$

It follows that, if the goal is the maximization of the objective function $w(\boldsymbol{q})$, then k_0 must be defined positively. On the other hand, it shall be negative if what is needed is the minimization of $w(\boldsymbol{q})$

The task space velocities may be expressed by expressing the vector $\dot{\boldsymbol{q}}_{\boldsymbol{opt}}$ using the generalization defined in Eq. (3.4); after that, the task priority framework can be employed by replacing it in Eq. (3.8). In this circumstance, the main task is being carried out (often the end-effector following a commanded reference trajectory), while the auxiliary tasks are performed without interfering with the higher priority tasks. It can be easily seen at this point that by increasing the number of redundancy degrees, it is possible to introduce more constraints or secondary tasks to the robotic system.

3.2.4 Case study

This section aims to demonstrate how the methodology that has been outlined in the previous section has been applied to address a particular real-world issue, which has been briefly outlined in the introductory section. Specifically, taking into account the case in which three mobile robots are placed on a flat surface that contains static objects. The kinematics of the mobile robots has been chosen to be omnidirectional. Additionally, each of the mobile robots is attached to the next one by a slack tether. These robotic systems are designed to possess a safety radius equal to 0.5 meters. Furthermore, the first robot is attached to the "ground" by means of the first tether, whereas the last robot is just connected to the second by a tether. Finally, the winch on each mobile robot controls the cable by applying a steady force of tension. The tether is supposed to provide either power or medium to each mobile robot. Figure 3.2a illustrates the just described system as well as the environment, whereas from Figure 3.2b it is evident the analogy between the considered chain of TMRs and a planar RM for this case scenario.

The objectives that are highly desired to achieve within this case study are summarized as follows:

- The mobile robots must move inside an environment that comprises obstacles; the leader robot must be able to reach the desired commanded target point in the task space by following a computed beforehand reference trajectory;

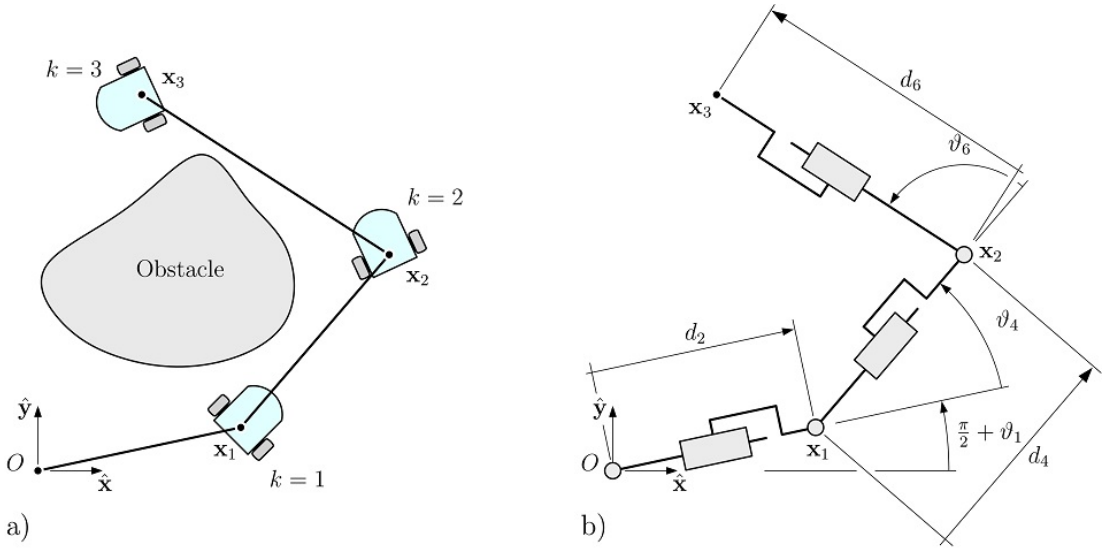


Figure 3.2: (**a**) Top view of three TMRs surrounding an obstacle; (**b**) the same robotic system viewed as a planar RM [329].

- The mobile robots must avoid colliding with each other as well as the obstacles;
- Neither of the three cables is allowed to both touch the ground and come into collision with any of the nearby obstacles;
- The distance between each subsequent robot must be determined by balancing local minimization and uniformity with the others' tethers.

The parameters related to the Denavit-Hartenberg (DH) convention are presented in Table 3.1 and are based on the diagrams shown in Figure 3.2b. Note that by defining the joints this way, and according to the DH convention, the system has three zero-length links, i.e. two consecutive joints are coincident; specifically the prismatic joint with the subsequent revolute joint.

Table 3.1: Denavit-Hartenberg parameters table for the robotic manipulator described in Figure 3.2a [329].

Link	$\boldsymbol{a}$	$\boldsymbol{d}$	$\boldsymbol{\alpha}$	$\boldsymbol{\theta}$
1	0	0	$\pi/2$	$\pi/2 + \vartheta_1$
2	0	d_2	$-\pi/2$	0
3	0	0	$\pi/2$	ϑ_3
4	0	d_4	$-\pi/2$	0
5	0	0	$\pi/2$	ϑ_5
6	0	d_6	$-\pi/2$	0

As can be easily seen, the manipulator possesses $n = 6$ degrees of freedom. If the tuple coordinates (x, y) and orientation ϕ of the end-effector are provided in relation to a fixed reference frame, a generic task defined in the bi-dimensional

space is entirely described. That means that for the equivalent manipulator described above these considerations lead to $r = 3$. Additionally, functional redundancy can be leveraged to get $r = 4$ if the task is just to place the end-effector without consideration of its orientation.

3.2.5 Tasks definition

For the considered system, which for the moment is unconstrained, additional restrictions should be established in order to accomplish the goals that have been previously outlined. This may be accomplished by introducing several suitable secondary tasks and by utilizing the task priority framework, that has been reported in Sec. 3.2.3. This section defines and illustrates the main task together with the auxiliary tasks that will be used in the use case scenario. More precisely, Eq. (3.8) and Eq. (3.10) are used in order to use the Jacobians described in the next paragraphs.

Primary Task: Trajectory tracking

The equivalent RM's end-effector capability to track a computed beforehand reference trajectory is considered to be its primary assigned task. It follows that Eq. (3.4) can be utilized in order to define the main task as follows,

$$\dot{\boldsymbol{x}}_{\boldsymbol{e}} = \boldsymbol{J}_{\boldsymbol{e}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.12}$$

where $\boldsymbol{J}_{\boldsymbol{e}}(\boldsymbol{q}) \in \mathbb{R}^{2\times 6}$ is the end-effector's associated Jacobian and $\dot{\boldsymbol{x}}_{\boldsymbol{e}} \in \mathbb{R}^2$ is the end-effector's task velocity vector, which is therefore defined as follows,

$$\boldsymbol{J}_{\boldsymbol{e}}(\boldsymbol{q}) = [J_1, J_2, \ldots, J_6] \tag{3.13}$$

Hence, if supposing that $\boldsymbol{p}_p^0$ and $\boldsymbol{p}_k^0$, both expressed into the base reference frame, are, respectively, the position vectors defining a generic point $\boldsymbol{p}_p$ and the k-th joint, it follows that:

$$J_k = \begin{cases} \Gamma \cdot \Omega_k \cdot \Lambda & \text{if } k\text{-th prismatic} \quad (3.14) \\ \Gamma \cdot \left[\Omega_k \times (\boldsymbol{p}_{\boldsymbol{p}}^{\boldsymbol{0}} - \boldsymbol{p}_{\boldsymbol{k}}^{\boldsymbol{0}})\right] \cdot \Lambda & \text{if } k\text{-th revolute} \quad (3.15) \end{cases}$$

with

$$\Omega_k = \prod_{i=1}^{k} T_i^{i-1}, \quad \Lambda = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T, \quad \Gamma = \begin{bmatrix} \Lambda & \Lambda \end{bmatrix}^T \tag{3.16}$$

If the end-effector's associated Jacobian needs to be determined, it follows that the end-effector's position vector $\boldsymbol{p}_e^0$ should be substituted instead of the generic vector $\boldsymbol{p}_p^0$.

For what concerns the secondary tasks, the ones that will be used are: *joints limit avoidance* and *obstacle avoidance*. The detailed implementations are described in the following paragraphs.

Secondary Task: Joint limits avoidance

The first additional task that will be assigned is referred to as joint limits avoidance, which has a lesser priority than the aforementioned main task. The associated objective function may be defined as follows and thus the GPM approach will be leveraged [330]:

$$w(\boldsymbol{q}) = -\frac{1}{2n}\sum_{i=1}^{n}\frac{(q_i - q_{i,t})^2}{(q_{i,M} - q_{i,m})} \tag{3.17}$$

where q_i denotes the joint variable associated with the i-th joint, and $q_{i,t}$, $q_{i,M}$ and $q_{i,m}$ are the i-th joint's desired, maximum, and minimum values, respectively. In this instance, the revolute joints are unrestricted whereas the prismatic ones are. More specifically, the maximum joint value is determined by resolving the catenary problem, which entails determining the maximum separation between the mobile robots that ensures that the cable does not touch the ground. On the other hand, the minimum joint value has been defined as a function of the safety circles associated with two subsequent mobile robots and therefore is related directly to the physical dimensions of the mobile robots.

Subsequently, Eq. (3.11) may be used to derive the relation $\dot{\boldsymbol{q}}_{\boldsymbol{0}} = \boldsymbol{K}\hat{\boldsymbol{\varphi}}$, where $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is defined as the gain diagonal matrix; on the other hand $\hat{\boldsymbol{\varphi}} \in \mathbb{R}^6$ is a vector in which its i-th component is specified as $\hat{\varphi}_i = (q_{i,t} - q_i), \quad \forall i = 1, \ldots, n$. It must be noted that this has been obtained by substituting the constant gain k_0 with a diagonal gain matrix $\boldsymbol{K}$. Specifically, this matrix will be used to prioritize a particular joint variable in relation to the others. The $\boldsymbol{K}_{i,i}$ elements are made up of the values that have been obtained by differentiating Eq. (3.17). Choosing a vector $\dot{\boldsymbol{q}}_0 \propto \nabla w(\boldsymbol{q})$ minimizes the term $(q_{i,t} - q_i)$. The reason for setting the coefficients $\boldsymbol{K}_{i,i} = 0$ for $i = 1, 3, 5$ is because the revolute joints are unrestricted in their domain, thus $q_i \in [-\pi, \pi]$, with $i = 1, 3, 5$. It is worth pointing out that, by using a gain matrix it is possible to specify different gains on the diagonal in order to give more importance to a joint with respect to the others.

This task's $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{2,6}$ corresponds to the one associated to the end-effector given in Eq. (3.13). Specifically, where to compute its various terms, Eq. (3.14) and Eq. (3.15) must be used; the position vector $\boldsymbol{p}_e^0$ relative to the base frame must be substituted in $\boldsymbol{p}_p^0$. By looking at the size of the Jacobian, it is possible to say that the introduction of this task reduces the number of DOFs of the equivalent manipulator by a factor of two.

Secondary task: Obstacle avoidance

Some control is required to ensure that the mobile robots can traverse the task space securely and also to prevent them from running into any obstacles. Additionally, this may be accomplished by utilizing a certain amount of free DORs and afterward, as done in the preceding paragraph, establishing an additional task that has the role to provide the system for obstacle avoidance capabilities.

Considering that the end-effector, i.e. the leader mobile robot, is already been moving along a path that is computed beforehand to be free from obstacles,

then the addition of this secondary task permits the system to generate internal motions in order to rearrange the structure to ensure that the internal links will not run into obstructions.

Every time step requires the definition of a critical point $\boldsymbol{x_c}$, as stated in [317]. This is referred to as the location on the manipulator that is closest to the obstacle. Therefore, it follows that this task may be expressed differently as follows,

$$\dot{\boldsymbol{x}}_{\boldsymbol{c}} = \boldsymbol{J_c}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.18}$$

where $\boldsymbol{J_c}(\boldsymbol{q}) \in \mathbb{R}^{2\times 6}$ is defined as the Jacobian matrix associated with the critical point, while $\dot{\boldsymbol{x}}_{\boldsymbol{c}} \in \mathbb{R}^2$ is the evasive speed vector. The Jacobian $\boldsymbol{J_c}(\boldsymbol{q})$ associated with this newly introduced point depends on the appropriate link to which this point belongs since it could be located anywhere along the manipulator's chain. It follows that:

$$J_c = \begin{cases} \begin{bmatrix} J_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \text{if} \quad \boldsymbol{x_c} \in \text{Link}_1 \\ \begin{bmatrix} J_1 & J_2 & J_3 & 0 & 0 & 0 \end{bmatrix} & \text{if} \quad \boldsymbol{x_c} \in \text{Link}_2 \\ \begin{bmatrix} J_1 & J_2 & J_3 & J_4 & J_5 & 0 \end{bmatrix} & \text{if} \quad \boldsymbol{x_c} \in \text{Link}_3 \end{cases} \tag{3.19}$$

where J_k, with $k = 1 \dots 5$, is calculated again by using Eq. (3.14) and Eq. (3.15), hence the vector $\boldsymbol{p_p^0}$ must be replaced with the position vector $\boldsymbol{x_c}$.

There are many methods to define the evasion speed vector $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, e.g. as a function of the distance, a blending function, or a constant module vector.

The vector representing the evasion speed, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, is oriented in such a way that maximizes the separation between both the critical point and the nearest obstacle. As a result of this consideration, is that it is possible to establish a scalar objective function, that if maximized will fulfill this task, and is defined as follows,

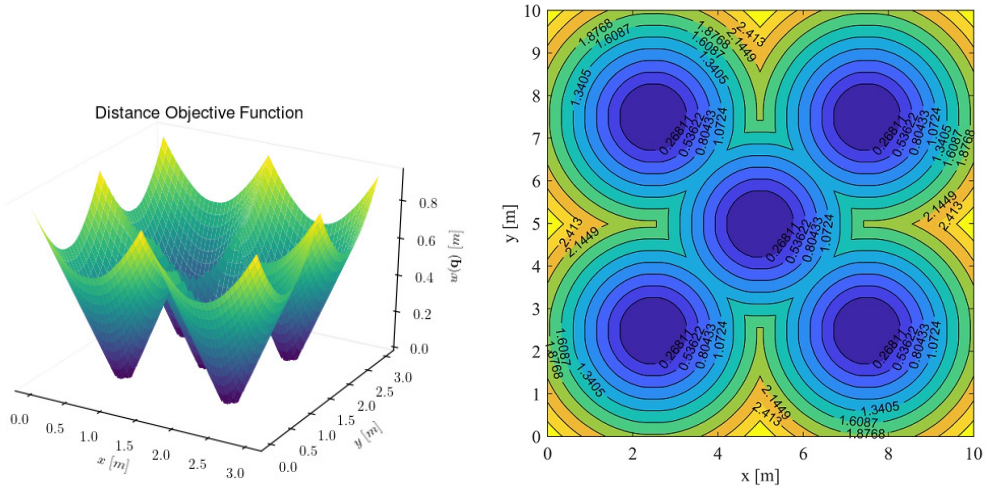
$$w(\boldsymbol{q}) = ||\boldsymbol{d_c} - \boldsymbol{x_c}|| \tag{3.20}$$

where $\boldsymbol{d_c}$ is defined as the closest obstacle's position vector. It is apparent that $\boldsymbol{d_c}$ and $\boldsymbol{x_c}$ must be defined with respect to the same coordinate system. An example of this objective function $w(\boldsymbol{q})$, can be seen from Figure 3.3a. However, within this investigation it has been decided not to utilize this function directly in order to define this additional task; rather it has been elected to use its normalized gradient. In addition to this, it has been defined a nominal evasion speed, together with an activation function, as indicated in the following.

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{c}} = v_0 \dfrac{\nabla w(\boldsymbol{q})}{||\nabla w(\boldsymbol{q})||} & ||\boldsymbol{d_c} - \boldsymbol{x_c}|| < d_{min} \\ \dot{\boldsymbol{x}}_{\boldsymbol{c}} = \boldsymbol{0} & ||\boldsymbol{d_c} - \boldsymbol{x_c}|| \geq d_{min} + \varepsilon \end{cases} \tag{3.21}$$

wherein d_{min} is a tunable parameter representing the safety distance that will force this secondary task to activate. The well-known *Zeno Phenomenon* has been successfully avoided or at least minimized by the introduction of $\varepsilon > 0$; this has the effect of increasing the stability of the controller and minimizing system

oscillations. Moreover, in MATLAB, a discrete two-dimensional grid reflecting the workplace environment has been created. This has been created by utilizing the logical grids, namely, Occupancy Grids, where each occupied cell is set to true while if not occupied is set to false. Subsequently, over the just defined grid has been defined the scalar objective function $w(\boldsymbol{q})$. Specifically, to each grid cell, a value has been set which then represents the minimal Euclidean distance computed from the current grid cell and the closest in which an obstacle is present. A cell's allocated distance value has been set to zero if there is an obstruction present. The gradient has then been computed and mapped across the same grid in preparation for future demands. Figure 3.3b clearly demonstrates that the direction of the gradient is perpendicular to the level surfaces.



(a) (b)

Figure 3.3: In (a) a sample objective function $w(\boldsymbol{q})$ used for the obstacle avoidance sub task in three dimensions; in (b) the view of its level surfaces [329].

It is important to point out that the formulation of $w(\boldsymbol{q})$ has two key features. The first is that there are no limitations on representing convex or regular geometries due to its mapping procedure across a discrete grid, thus making it feasible to describe any sort of obstacle. The second, instead, is that any real points cloud-based systems, such as ultrasonic, LiDAR, or any other sensor, could easily be integrated within this formulation.

Although the escape velocity v_0 can be at start assumed to be constant, within this work this speed has been specified explicitly as being inversely proportional to the critical separation $\boldsymbol{d_c}$ in order to achieve a smoother behavior as follows,

$$v_0 = \frac{k_{v_0}}{||\boldsymbol{d_c} - \boldsymbol{x_c}||} \tag{3.22}$$

As seen for the joint limits avoidance secondary task in the previous paragraph, the introduction of this task has the effect of reducing by two the number of degrees of freedom. It is possible therefore to reduce this number from two to

one using the versor of the gradient, which has been defined above. Therefore, it is possible to leverage a new Jacobian $\boldsymbol{J_n}(\boldsymbol{q}) \in \mathbb{R}^{1\times 6}$ associated with the obstacle avoidance secondary task, which is therefore defined as follows,

$$\boldsymbol{J_n}(\boldsymbol{q}) = \boldsymbol{n_c}^T \boldsymbol{J_c}(\boldsymbol{q}) \tag{3.23}$$

where,

$$\boldsymbol{n_c} = \frac{\nabla w(\boldsymbol{q})}{||\nabla w(\boldsymbol{q})||} \tag{3.24}$$

3.2.6 Control

The control architecture which has been utilized to fulfill the objectives given in the previous section will be outlined throughout this paragraph, along with the relevant control flow. The CLIK (Closed Loop Inverse Kinematic) method, already highly used for serial robotic manipulators, as introduced in [312], serves as the foundation for the controller used for controlling the chain of TMRs. The vector of the tracking error $\boldsymbol{e}(t)$ associated with the end-effector, i.e. the leader mobile robot, at time t must first be introduced:

$$\boldsymbol{e}(t) = \boldsymbol{x_d}(t) - \boldsymbol{x_e}(t) \tag{3.25}$$

where, at time t, the term $\boldsymbol{x_e}(t)$ represents the vector defining the position of the end-effector and $\boldsymbol{x_d}(t)$ represents the location of the desired point on the trajectory, i.e. the location where the end-effector should be. Eq. (3.25) can be differentiated, and by utilizing Eq. (3.4), it is possible to obtain the following:

$$\dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) - \dot{\boldsymbol{x}}_{\boldsymbol{e}}(t) = \dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) - \boldsymbol{J_e}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.26}$$

Therefore, it has been chosen the following equivalent system,

$$\dot{\boldsymbol{e}}(t) + \boldsymbol{K}\boldsymbol{e}(t) = 0 \tag{3.27}$$

It had been demonstrated that the aforementioned equivalent linear system, by using gain matrices $\boldsymbol{K}$ which is positively defined, is characterized by being asymptotically stable. Therefore, the algorithmic solution for the IK for the task of the end-effector, or the leader mobile robot, tracking a reference trajectory is defined as follows,

$$\dot{\boldsymbol{q}} = \boldsymbol{J_e^{\#}}(\boldsymbol{q})(\dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) + \boldsymbol{K}\boldsymbol{e}(t)) \tag{3.28}$$

Since the main task has been identified as the trajectory tracking one, it follows that Eq. (3.28) may be used in place of Eq. (3.8). Moreover, it is possible to derive the algorithmic solution for the multi-task RM by also replacing the additional tasks that have been specified in the preceding section. The vector $\boldsymbol{q}$ of the joint variables is therefore given via time integration. Finally, by utilizing Eq. (3.8), it is possible then to conclude that:

$$\boldsymbol{q}(t) = \int_{t_0}^{t} \boldsymbol{J_e^{\#}}(\boldsymbol{q})(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{e})dt + \int_{t_0}^{t} \left(\sum_{i=2}^{t} \boldsymbol{P_{i-1}^{A}}(\boldsymbol{q}) \boldsymbol{J_i^{\#}}(\boldsymbol{q}) \dot{\boldsymbol{x}}_{\boldsymbol{i}} \right) dt \tag{3.29}$$

where t, again, represent the number of tasks, and for t equals to 1, the outcome is equal of Eq. (3.28).

It is apparent that Eq. (3.29), for numerical simulation purposes, must be time discretized. It follows then that the integrals symbols must be replaced with a summation, as well as a suitable sample period must be selected because the time integration must be performed numerically. Hence, the right choice of this parameter will help on avoiding numerical instabilities.

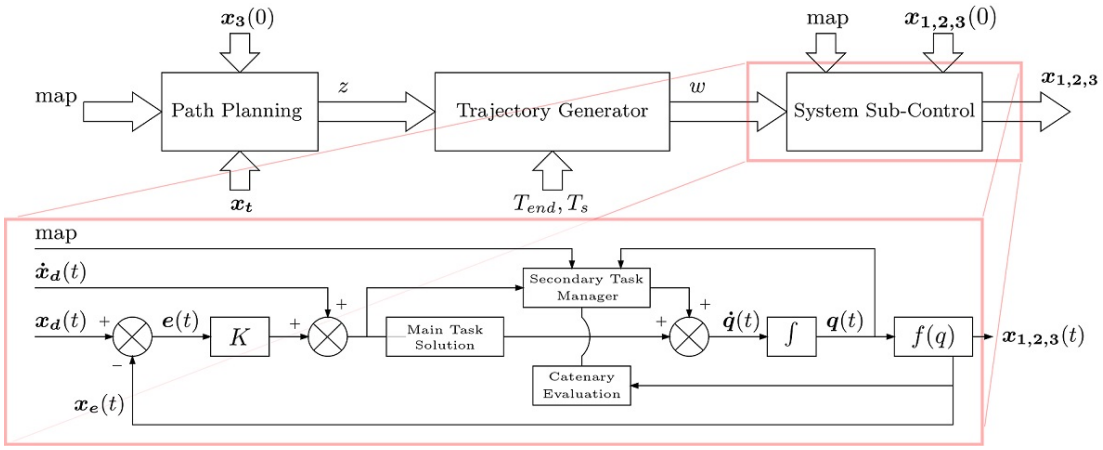


Figure 3.4: At the top, the logical scheme of how the overall algorithm works, in which are reported the three main blocks; at the bottom, the expansion of the System Sub-Control Block containing the CLIK algorithm which evaluates the shape of the catenaries and the secondary tasks' resolution [329].

"Centralized Control" is the type of control that will be used within this investigation. It is important to point out that the controller, that will be introduced, will assume a "fleet manager" behavior. Three distinct primary blocks may be found in the control architecture, as illustrated in Figure 3.4, and therefore are listed and briefly discussed below.

I. *Planner*: This block is in charge of the task of creating a suitable path, which is free from obstacles, for the equivalent RM's end-effector. This block takes in input an existing environment's discrete map (the occupancy grid defined in the previous section) and the equivalent RM's initial position. The RRT* [331] graph search technique has been leveraged in order to get the free obstacle path, which is successively saved in a data structure referred here to as $\boldsymbol{z}$. The effect of this block can be seen graphically from Figure 3.5(a).

II. *Trajectory generator*: With reference to Figure 3.4, this block is directly connected with the output of the planner from which it receives the free obstacle path data structure $\boldsymbol{z}$, and it's connected to the block *System Sub-Control*. This block is in charge of creating a trajectory that traverses all the

points that have been produced by the previous block given a required time, in which the trajectory should be completed. Additionally, within this block, a B-spline curve has been utilized in order to create a time-based trajectory. Specifically, for the use case taken in this investigation, a value $T_{end} = 15s$ has indeed been set as the trajectory's duration, whereas $T_s = 0.05s$ has been set as the sampling interval. The block outputs a trajectory data structure $\boldsymbol{w} = [\boldsymbol{X}, \dot{\boldsymbol{X}}, \ddot{\boldsymbol{X}}] = f(T_{end}, T_s, \boldsymbol{z})$ where its components are the time-series of the generated values $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ accordingly, where $t = 0, \ldots, T_{end}$. These values represent respectively the position, the speed, and the acceleration associated with the trajectory points, and that the end-effector should possess while traversing them. The length of these single data structure's components is $N = T_{end}/T_s$ points. The effect of this block, for a sample environment, can be seen graphically in Figure 3.5(b).

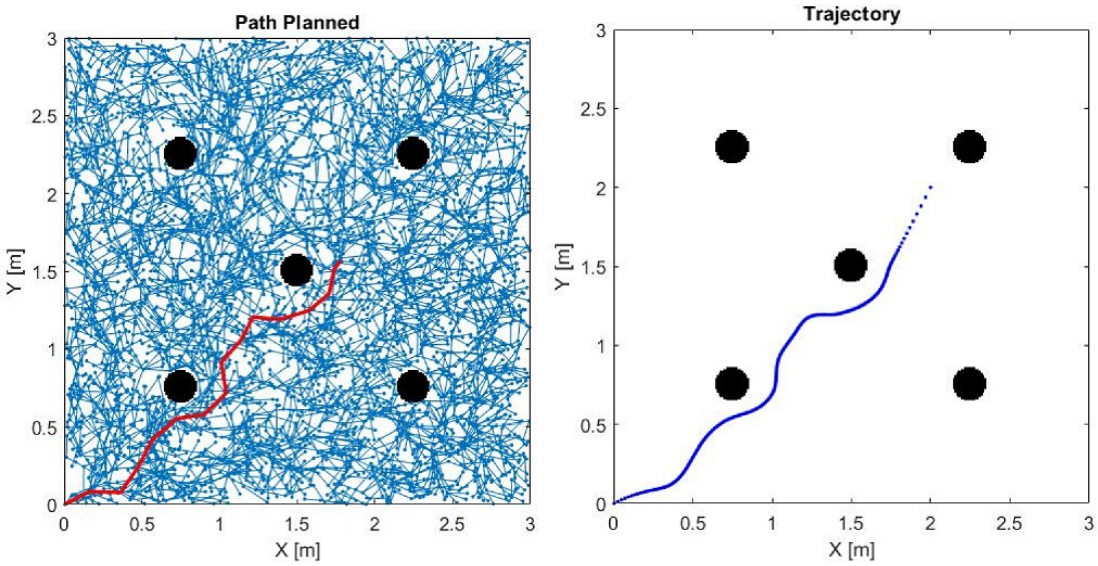
III. *System Sub-Control*: The CLIK algorithm operates in this block, making it the most crucial one. It accepts as entry the data structure of trajectory $\boldsymbol{w}$ provided by the *Trajectory Generator* as well as another data structure providing essential map information (for instance the distance field). Subsequently, this block outputs a data structure that contains the single trajectories that will be separately assigned to the single mobile robot constituting the chain of TMRs, and which need to track. These trajectories are shown in Figure 3.4 and denoted with $x_{1,2,3}$. Furthermore, this block may be expanded to reveal a subsystem control loop which in turn contains three major components:

- *Main CLIK Algorithm*: This block has the role to compute the RM's IK and provides a reference signal to the equivalent manipulator's end-effector, which in turn must track it;
- *Secondary Tasks Manager*: The single elements employed in the multi-task framework are calculated and arranged within this block;
- *Parallel Catenary Evaluation*: This block verifies the precise attitude and the mapping of the catenary-based cables at each iteration. Additionally, in order to control the elongation of the tether, it is also responsible for rearranging the matrix of the gains associated with the secondary task of joints limits avoidance.

The Secondary Task Manager has the two unique states listed below and is modeled as a finite state automaton (FSA) with two states:

State A. Only the principal task and the joints' limit avoidance are triggered when all of the links of the manipulator are outside of the hazard collision region;

State B. The additional task associated with the obstacle avoidance activates whenever any point of any equivalent manipulator link crosses into the danger zone, taking priority over the joint limits avoidance additional task, while the latter's priority is dropped to the lowest priority. This



(a) (b)

Figure 3.5: Sample environment: In (a) the resulting graph and the feasible path obtained by the planner block; in (b) the time-based trajectory synthesized by the trajectory generator block.

results in giving up some of the ability to minimize the separation between every mobile robot while obtaining increased maneuverability around obstacles. Until any of the colliding points is still inside the danger zone, this condition continues to be in operation. The danger area required to deactivate this secondary task is the original danger area summed by a ε in order to avoid the Zeno phenomenon as discussed in the previous sections.

It must be pointed out that Eq. (3.2) has been utilized at every iteration of the main control loop. This has been done in order to map the joints' position of the equivalent robotic arm - the single mobile robots - starting from its joint variables. This process has the effect of generating the three different trajectories assigned to the single mobile robots. This process can be seen from the diagram in Figure 3.4.

Parallel catenary evaluation

The tethers' behavior and their shape must be calculated and described in conjunction with the control of the equivalent RM since they are treated as prismatic joints, such that it is possible to exploit the presented kinematic analogy between chains of TMRs and this equivalent robotic arm. In particular, this is carried out to keep the tether from contacting the ground, which is in contrast with the requirements of the use case scenario.

Spline, catenary, and parabola curves can be used to model and simulate flexible tethers [332]. The catenary depends on the tension and weight of the cable, as well as the suspending points' elevation. In the following is reported the

parametric equation for catenary-based curves [333]:

$$y = c\left(\cosh\left(\frac{x}{c}\right) - 1\right) \tag{3.30}$$

where,

$$c = \frac{T_0}{\rho g} \tag{3.31}$$

The variables T_0, ρ, and g represent the cable's horizontal tension force, the cable's specific weight, and the gravitational acceleration, respectively. The reference frame used for the curve parameterization is centered at the curve's lowest, i.e. the tuple (x, y). An example of a tether suspended between two anchor points, having the same height, is illustrated in Figure 3.6.

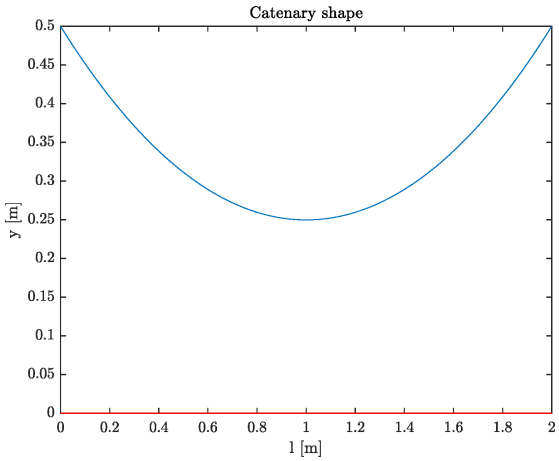


Figure 3.6: Example of a suspended tether fixed at two points with same height

Taking into account the pulley's torque resistance assumed to be $T_r = 1.35\,\mathrm{N\,m}$, the pulley radius $r = 0.1\,\mathrm{m}$, as well as the robots' anchoring point elevation $H = 0.43\,\mathrm{m}$. By utilizing Eq. (3.30) and assuming a cable-specific weight equals to $\rho = 0.15\,\mathrm{kg\,m^{-1}}$, it can be calculated that $d_{max} = 5.84\,\mathrm{m}$ is the greatest horizontal separation that can be maintained between two successive mobile robots of the TRM chain, while still ensuring that the tether does not come in contact with the ground.

The minimum and the maximum distances which have been computed become the prismatic joints limits in the joints limits avoidance secondary task.

Since the catenary depends only on geometrical parameters and the tension of the cable which can be considered constant, the shape of the catenary has been evaluated previously and independently from the control algorithm. This allows the mapping of the shape and creates an interpolation that maps the distance between two robots to the cable sag. It follows that this will produce benefits from a computational point of view in the control loop algorithm since the catenary won't be computed at each iteration.

For every value of the tether's sag—or equivalently, the robots' horizontal separation—a value has to be mapped for the gain matrix $\boldsymbol{K}$, used for the additional task of the joints limits avoidance, must be chosen based on the formulae reported

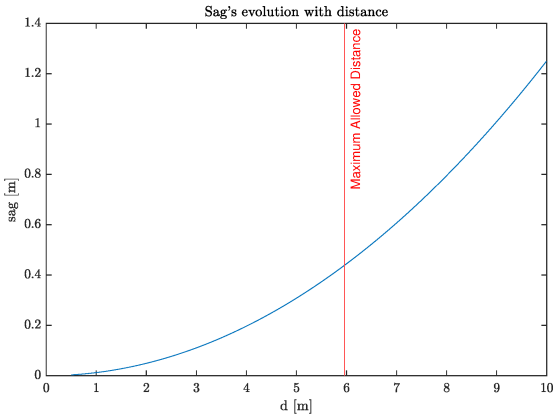


Figure 3.7: Catenary's sag evolution over the horizontal distance between two hanging points.

in Eq. (3.30). In order to accomplish this, a proper function has been designed and constructed as follows using the combination of two exponential functions as well as a constant gain G:

$$\boldsymbol{K}_{i,i}(q_i) = a_1 e^{b_1 q_i} + a_2 e^{b_2 q_i} + G, \qquad i = 2, 4, 6 \tag{3.32}$$

The equation that has been reported above requires the robot's horizontal distance as input. The goal of the exponential functions is to gradually increase the gains employed in the joint limits' secondary avoidance task under specific circumstances. This, as an instance, occurs if either the horizontal spacing of the robots exceeds the minimum safety distance d_{min}, which has been introduced to ensure that they will not collide, or when it happens that the maximum distance exceeds d_{max}, which has been defined as distance beyond which the cable will go in contact with the ground.

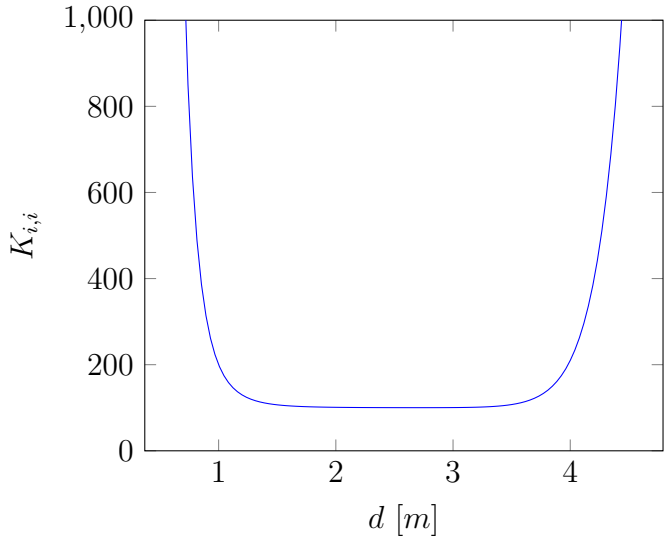


Figure 3.8: Joint's limit avoidance gain's trend with respect to robot's horizontal distance

Parameters tuning, performance function definition and optimization

Numerous system characteristics, including the various introduced gains as well as the evasion speed, must be adjusted in order to create a worldwide smooth behavior and minimal energy loss. To compare individual simulations, it is also important to establish performance functions or objective functions. The description of the simulation environment, i.e. the logical map, must be the same for every conducted simulation in order to give coherent results.

Within this paragraph, all the objective functions that have been used in the scope of this investigation are listed. Moreover, their values are discretized into N equally spaced samples with $k = 0, \ldots, N$ as their index. Moreover, it has been elected that all the performance functions must be minimized at the same time:

$$f_1 = \sum_{k=0}^{N} \dot{\boldsymbol{q}}(kT_s)^{\boldsymbol{T}} \dot{\boldsymbol{q}}(kT_s) \tag{3.33}$$

$$f_2 = \sum_{k=0}^{N} \left[(d_1 - d_2)^2 + (d_2 - d_3)^2 + (d_3 - d_1)^2\right] \tag{3.34}$$

$$f_3 = \sum_{k=1}^{N} ||\boldsymbol{x_e}(kT_s)) - \boldsymbol{x_d}(kT_s)||^2 \tag{3.35}$$

$$f_4 = \sum_{k=1}^{N} \dddot{\boldsymbol{q}}(kT_s)^T \dddot{\boldsymbol{q}}(kT_s) \tag{3.36}$$

Notably, the energy used by the fictitious motors of the equivalent RM is taken into account in Eq. (3.33). The homogeneity of the separation between the robots is taken into consideration in Eq. (3.34), wherein $d_i = f(kT_s)$ with $i = 1, \ldots, 3$ represent the discrete counterpart of q_{2i} (i.e. the elongation of the $2i$-th prismatic joint). Finally, it is desired to try to maintain a constant robot separation distance by achieving minimization of this function. Additionally, Eq. (3.35) measures the precision with which the end-effector tracks the assigned reference trajectory. Finally, the structure's overall jerk is quantified by Eq. (3.36), whose minimization it's apparent will result in a system's smoother motion.

A blind penalty is imposed making $f_i = +\infty$, if somehow the end-effector is incapable of ending successfully the simulation for any cause. Failures of this kind happen when big values cause the system to become unstable, whenever the robot's links run into objects, or if the minimum separation between robots is smaller than what is permitted. The multi-objective optimization with genetic algorithm process (MOGA) has been used, and the appropriate set of parameters was ultimately determined.

3.2.7 Results

Four alternative scenarios have been simulated once the characteristics have been properly tuned in the previous paragraph. These scenarios are listed as:

- **Scenario A**: a square space with five obstacles, each represented with a disk of a particular diameter; this scenario has been thought to mimic a room with some columns placed in it; in this scenario, the leader robot must place itself behind the column in the center of the room;

- **Scenario B**: a setting with a walkway with a high degree of curvature that serves as the exit from the room. The final robot's objective in this scenario is to make it to the end of the corridor;

- **Scenario C**: a square space with parts of the walls built to resemble a little labyrinth. Within this scenario, the robot leader should take up a position in the corner at the top right of the room;

- **Scenario D**: a square room with a replica of a Cessna C-172 airplane, which in turn has to be subjected to operations performed by the mobile robots, for instance, robotic painting in this case. In this scenario, the robot leader must maneuver all around the aircraft and actually take a position on the opposite side.

It is worth pointing out that before feeding the map to the planner block, the obstacles' size has been scaled appropriately to accommodate for the physical characteristics of the robots. This has been done in order to achieve two features: introduce a margin factor and take care of the mobile robot's turning radius.

The equivalent manipulator postures are reported in Figure 3.9, for some simulations that have been performed. Specifically, these simulations have been performed on each of the four introduced scenarios, therefore the posture of the RM is assessed across a specific time step chosen equals to three seconds. The endpoints of the straight lines visible from the same figure represent the equivalent RM's revolute joints, or equivalently the mobile robots, while on the other end, the straight lines depict the links of the RM, or equivalently the tether, connecting the mobile robots composing the chain of TMRs, seen from above. It is evident that the system adjusts its structure to avoid obstacles close to it while also attempting to maintain adjacent mobile robots as close to one another as possible as well as reducing the horizontal distance between them.

Figure 3.10 displays the trajectories that the fleet controller has produced, and that the mobile robots then must track. The figure shows the generated trajectories for each of the introduced scenarios: scenario A is shown in the top row, scenario B is shown in the second row, scenario C is shown in the third row, and finally scenario D is shown in the bottom row. All other trajectories exhibit spikes caused by both numerical oscillations and the abrupt activation/deactivation of the additional tasks, with the sole exception of the leader mobile robot's trajectories, which, must be remembered, have been calculated beforehand at the planner level. As a consequence of this, trajectories assigned to the single mobile robots have been smoothed through a filtering process in order to get rid of the spikes. The paths in red represent the outcomes of this procedure.

Remember that the equivalent manipulator's q_2, q_4, and q_6 joint variables, which describe the elongation of the three prismatic joints, which according to

the kinematic analogy introduced also reflect the longitudinal distance between two adjacent mobile robots.

Four distance errors' temporal evolution are shown in Figure 3.11. At a given time t, the difference between both the mean of the horizontal separation of all mobile robots and their single horizontal separation is what they are regarded as. Additionally, as can be noted from the same figure, the errors remain nearly zero throughout the simulation, with the exception of a few moments which have been addressed as those moments in which the additional task of obstacle avoidance is triggered, leading to the RM arm trying to avoid obstacles while at the same time deactivating the additional task of the joints limits avoidance. Inaccuracies observed on the tracking rise as a result of this. It has been observed that this discrepancy, by considering the absolute value, is less than 0.008 m for scenario D, although it is still within 0.05 m for scenarios B and C. Last but not least, for scenario A, is smaller than 0.10 m.

The required instantaneous speeds for every mobile robot to fulfill the three trajectories that the controller has generated, specifically for scenario B, is shown in Figure 3.12. It must be pointed out that the velocities indicated have been normalized in relation to the nominal speed v_n, which is the required end-effector's (Robot 3) speed to follow the whole trajectory during the allotted simulation period. It can be noted that the spikes in the original trajectories result in significant instantaneous speeds and accelerations, on the other hand, it is also evident how the filtering of the trajectories has affected the data in this picture as well, i.e. by smoothing the instantaneous speeds. The red curve represents the filtered speed distribution that every mobile robot should assume to track the assigned reference trajectory.

Fitness: Sensitivity analysis

Within this subsection is provided a sensitivity analysis performed on two key parameters in order to understand the mechanisms underlying differences in the objective functions shown in Eqs. (3.33)–(3.36): the constant associated with the escape speed k_{v_0} as well as the minimum separation d_{min} which triggers the obstacle avoidance additional task. The decision to perform a sensitivity analysis on these parameters has been taken based on the fact that these have been observed as the two most important characteristics that are also grounded in physical quantities; in fact, the majority of other metrics are linked to the controller's gains and, as a result, poorly reflect the robot's high-level behavior. The allowed range of variation for the chosen parameters has been expressed by the formulas $k_{v_0} = k^*_{v_0} + [-50\%, +100\%]$, while $d_{min} = d^*_{min} + [-50\%, +100\%]$, where $k^*_{v_0} = 0.05\,\mathrm{m^2\,s^{-1}}$ and $d^*_{min} = 0.5\,\mathrm{m}$, respectively. The fitness functions display values between the following ranges: $f_1 \in [123.9412, 453.7501]\mathrm{J}$, $f_2 \in [0.0021, 0.6446]\mathrm{m^2}$, $f_3 \in [1.1405, 1.1407]\mathrm{m^2}$ and $f_4 \in [0.1559 \times 10^8, 2.7352 \times 10^8]\mathrm{m\,s^{-3}}$. For every one of the objective functions which have been mentioned in the previous section, the findings of this sensitivity analysis are shown in Figure 3.13 for each and will be discussed in the following.

The approach is successful in maintaining the spacing between the mobile robots uniform because, generally, the objective function f_2 stays relatively mod-

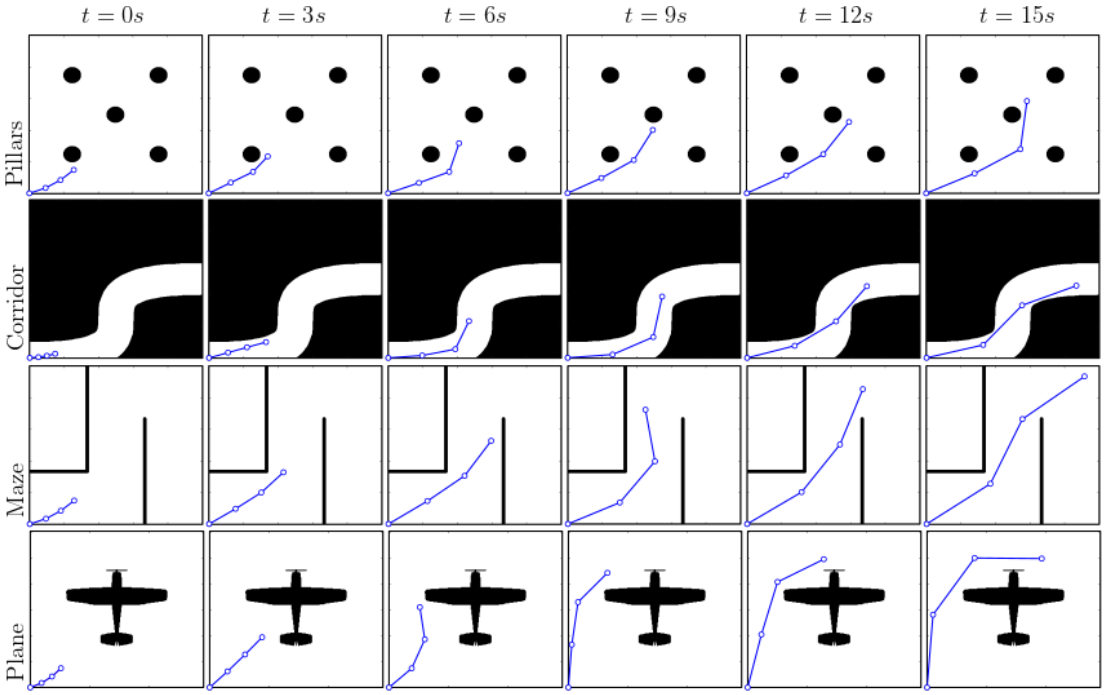


Figure 3.9: Evolution in time every three seconds of the pose of the equivalent manipulator. In the top row, the room with the five columns is shown (Scenario A); in the second row, the corridor is shown (Scenario B); in the third row, the maze is shown (Scenario C); and, in the bottom row, the room with the plane is illustrated (Scenario D) [329].

est in any configuration pair (k_{v_0}, d_{min}). This is generally correct for f_3, which demonstrates that with the great majority of settings, the robots' tracking error is maintained at a moderate level (k_{v_0}, d_{min}). The region highlighted in black depicts settings for the pair (k_{v_0}, d_{min}) which results in an impractical path planning, or one in which the robot runs into obstacles. Vertical bands may be observed on the objective functions' response surfaces; they look matched between the different charts and demonstrate some regularity between d_{min} values, or, in other words, that d_{min} appears to affect system performance in a step-like fashion. This is certainly not the case with k_{v_0}, which does in fact exhibit a very subtle impact.

The triggering of the additional task of obstacle avoidance, which imposes significant penalties on the system's overall performance, is the reason for these enormous discontinuities. In reality, the secondary tasks do not work following the lowest energy approach, which results in significant fitness losses upon activation, whereas instead, the primary task operates according to the lowest energy approach requirement (this can be easily seen by looking at the first component of Eq. (3.5) that indirectly introduces this behavior). Geometrically, the algorithm creates abrupt evasive motions roughly orthogonal to the overall direction of travel during obstacle avoidance maneuvers, which has a detrimental impact on the fitness function connected to energy. As a result, as anticipated, this is particularly evident in the cases of f_1 and f_4, or the ones related to energy and jerk.

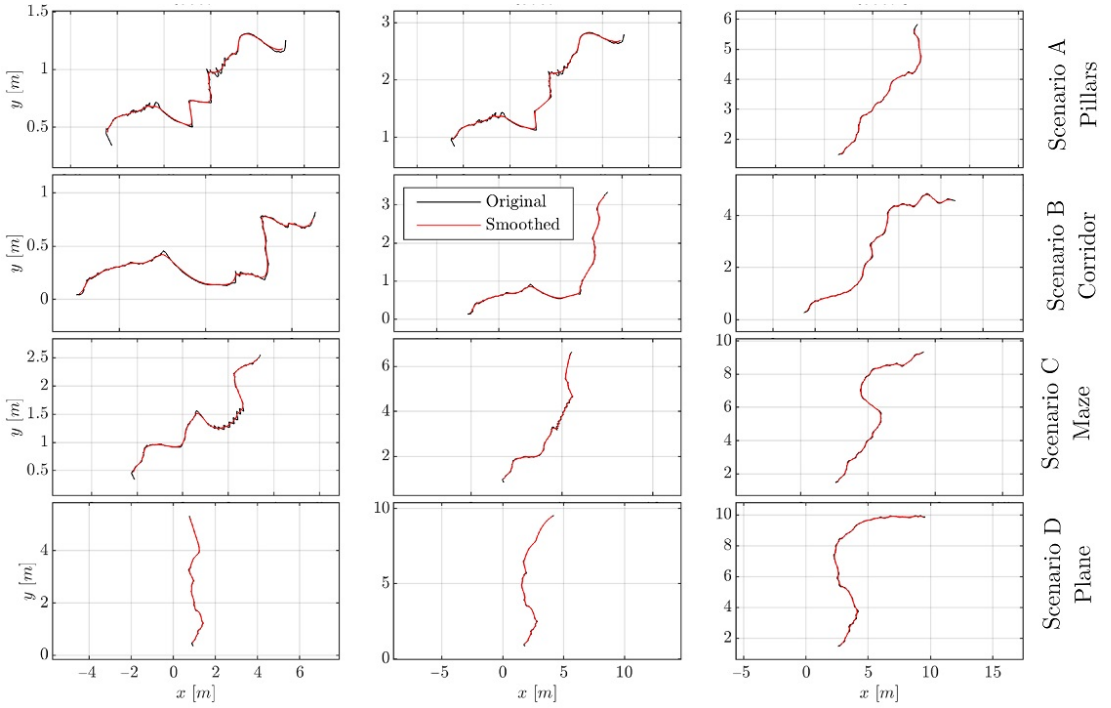


Figure 3.10: Original and filtered trajectories obtained for each of the mobile robots computed in the top row, the room with five columns (Scenario A); in the second row, the corridor environment (Scenario B); in the third row, the maze (Scenario C); and, in the bottom row, the room with the plane (Scenario D) [329].

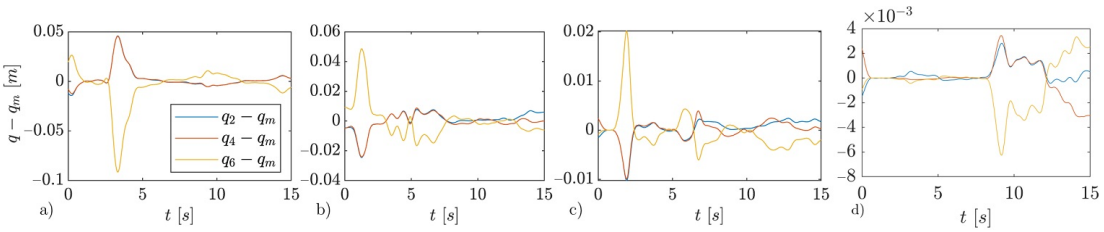


Figure 3.11: Distance error of the tree mobile robots evaluated in (**a**) the room with five columns environment (Scenario A); in (**b**), the Corridor (Scenario B); in (**c**), the Maze (Scenario C); and, in (**d**), the room with the plane (Scenario D) [329].

The difference between d_{min}'s and k_{v_0}'s contributions to f_i is likely due to the former's direct impact on the obstacle avoidance task's activation, i.e., when it activates, and the latter's only influence on how is this particular task's behavior.

The challenge of obtaining an optimal configuration (k_{v_0}, d_{min}) is evidently one of selecting both low values for k_{v_0} and d_{min} considering the morphology of the response surfaces of every objective functions. Nevertheless, it must be emphasized that this tends to lead to impractical setups (black areas indicated in the charts). As a result, if considering this as a multi-objective optimization problem, the boundary between the black and colored areas can be considered to be the Pareto front of the problem; as a result, configurations in this area have optimal values but minimal robustness due to their proximity to the infeasible

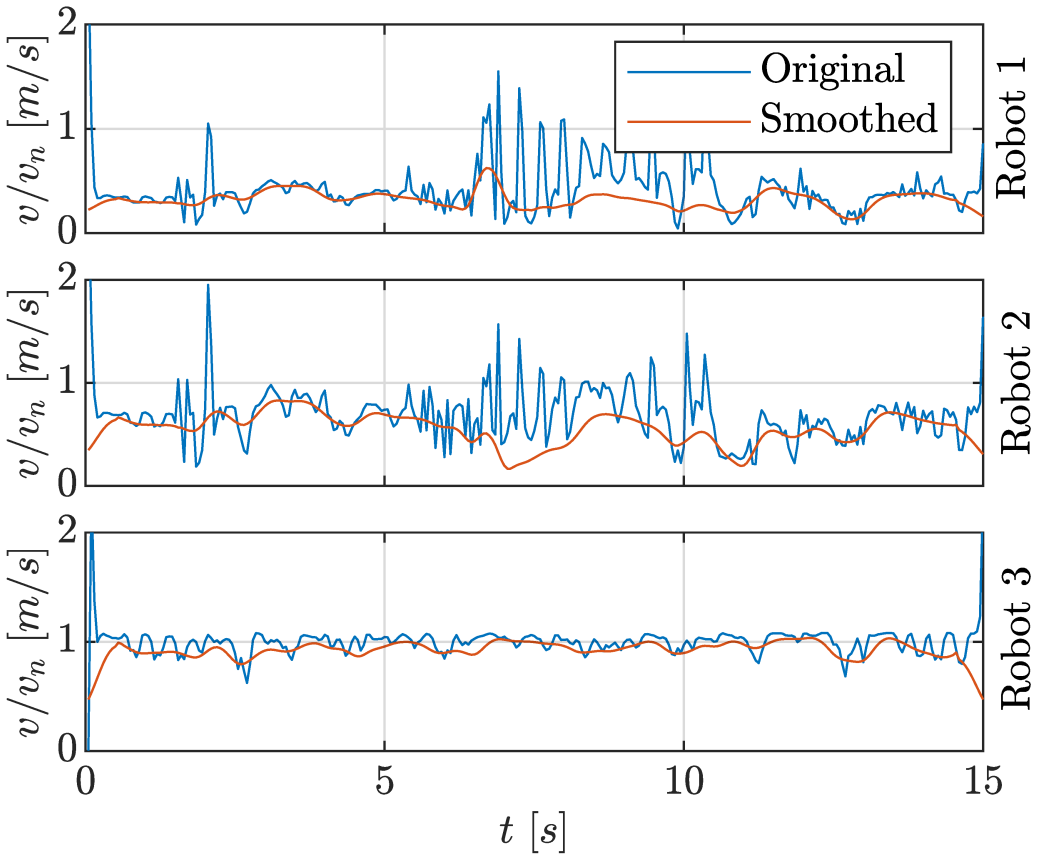


Figure 3.12: Normalized speed the mobile robots shall maintain in order to complete the assigned trajectory in scenario B [329].

region.

ROS and Gazebo simulation

Two simulations on Gazebo have been executed for scenarios C and D, respectively, in order to demonstrate the feasibility and validity of the methodology described in the preceding sections. Within this paragraph has been elected to use Gazebo in conjunction with the ROS-Gazebo bridge in order to simulate the robotic system in consideration. In particular, once the filtered trajectories are generated, they are stored in some convenient *.json* files which are then used as input for the simulation.

Two Python and XML-based ROS packages have been developed in order to perform a multi-robot simulation using Gazebo, which adds to the simulation as a whole. The first package is in charge of spawning three separate mobile robots inside the Gazebo simulator environment and initiating it depending on the map utilized in the other simulations. Three Neobotix MPO-500 mobile robots have been chosen for scenario C, and two MPO-500 and a MMO-500 for scenario D. All of them are omnidirectional drive mobile robots whose software is based on ROS, however, only the MMO-500 mobile robot does have an UR10 robotic arm mounted on top of it. The other package, on the other hand, is

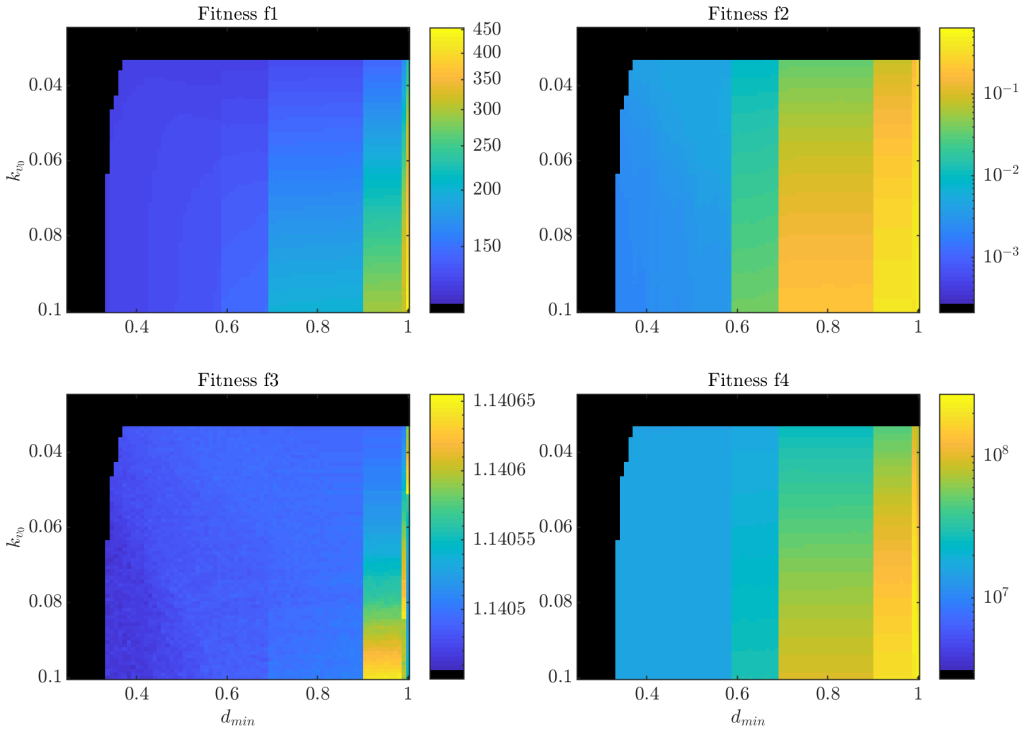


Figure 3.13: Sensitivity analysis of the fitness functions based on the variation of parameters k_{v_0} and d_{min} [329].

in charge of starting and establishing connections with as many ROS nodes as there are mobile robots in the environment. In addition to running a general PID controller for trajectory tracking and having a routine that stores the robot's internal variables into another *.json* file for data post-processing purposes, each of the nodes corresponds to a single process that is in charge of loading the *.json* file which includes the computed beforehand smoothed trajectories, generated by the fleet manager, and assigning the appropriate one to the mobile robot. It should be highlighted that every mobile robot has its own PID controller. This as a result translates into the advantages of being more computationally efficient and being coherent with the real-world scenario. This can be seen in graphical form in Figure 3.14, which represents the ROS graph displaying the nodes, their connections, and how they communicate via topics. A view of the simulation's last instant from scenario C is shown in Figure 3.15a. Figure 3.15b, which indicates the tracking error, instead displays three curves, one for every platform. 3.16 shows the same outcomes, but this time for scenario D.

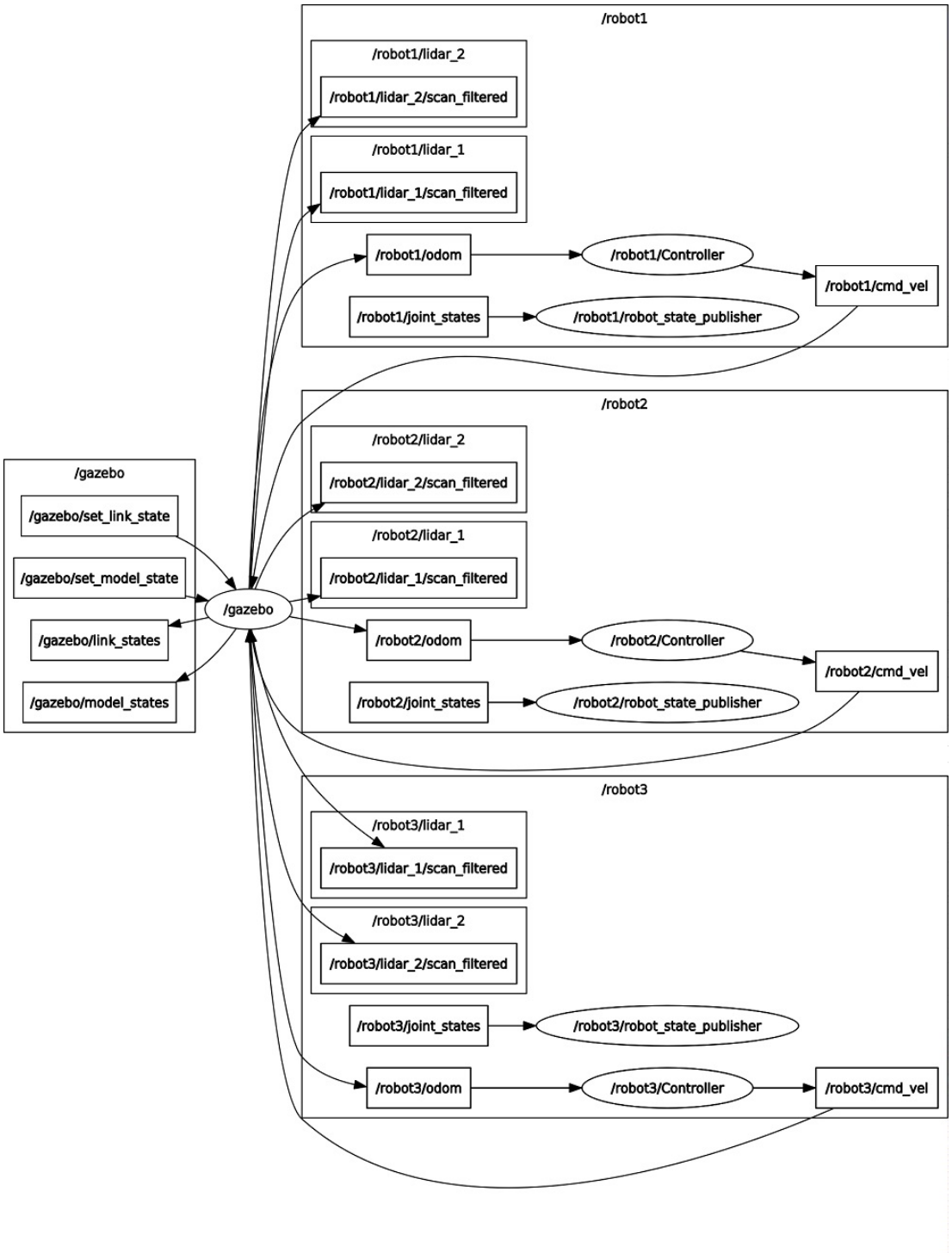


Figure 3.14: Rosgraph showing the ROS nodes communication diagram in the ROS application used for the tethered robot application simulation in Gazebo.

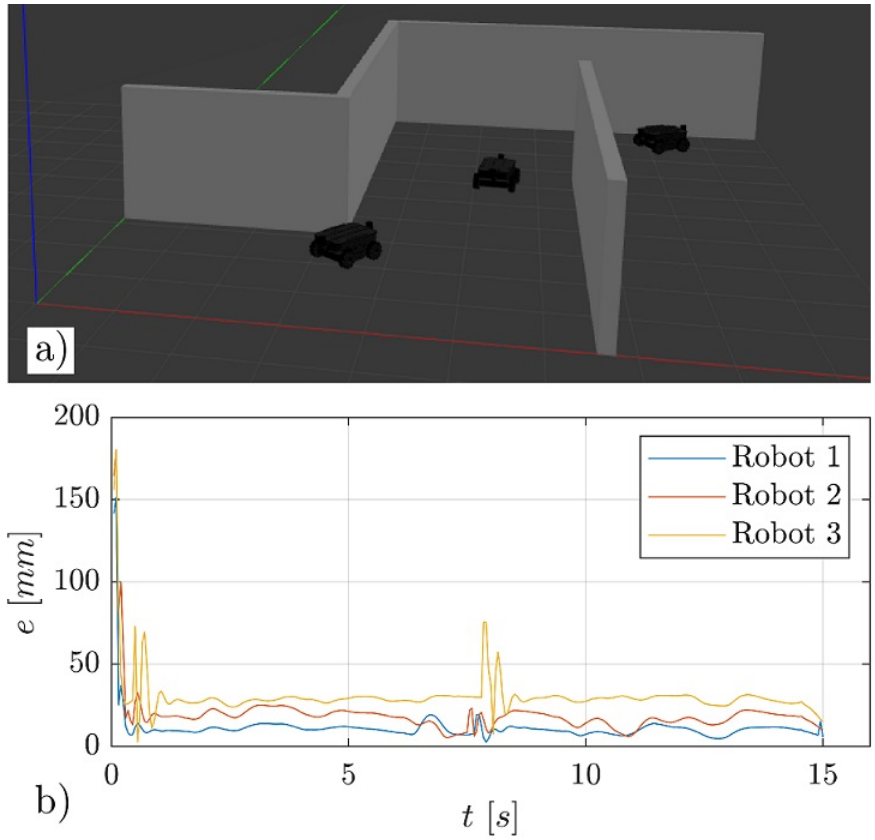


Figure 3.15: Gazebo simulation for the three omnidirectional robots in scenario C (Maze) (a) perspective view of the end of the simulation; (b) the tracking error [329].

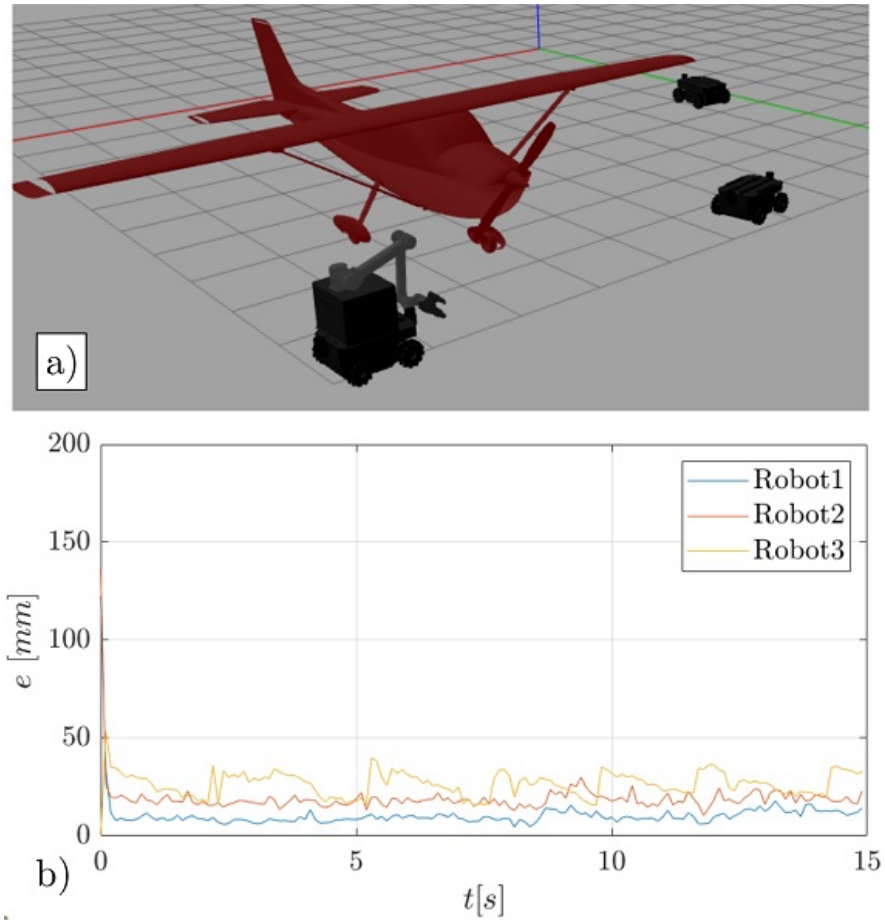


Figure 3.16: Gazebo simulation for the three omnidirectional robots in scenario D (Plane); (a) perspective view of the end of the simulation; (b) the tracking error [329].

Concluding it must be pointed out that all of the robots maintained a safe distance from one another and the obstacles placed in the environment throughout the simulations.

3.3 Crowd navigation

As discussed in the previous sections, the trends in industrial environments and Smart Factory settings are for mobile robots to be given more responsibilities and more tasks. Many of these tasks involve collaboration with human operators, while others only involve solely the mobile robot being aware of the presence of humans in the environment. It follows that the mobile robot must be "smart" enough to handle these situations and not be harmful to humans. Moreover, the mobile robot must be equipped with an intelligent controller that is aware of human presence and is responsible for resolving human-robot interactions, thus allowing the robot to be able to interact with humans. It should be emphasized that this requirement applies not only to mobile robots used in industrial and manufacturing settings but also to many common settings in the everyday context in which these systems are used. An example of these may be large environments such as airports or hospitals, where the mobile robot employed must be able to perform the task assigned to it, thus being able to navigate in a very crowded and chaotic environment.

This section aims to study and address one of the topics, which falls under the classification of tasks involving only the mobile robot's awareness of the presence of humans, and which is very interesting and attractive in the field of robot autonomy development. Specifically, this section is concerned with studying and addressing the problem of navigating a mobile robot within a dense moving crowd through the development of an "intelligent" controller that has been trained with reinforcement learning techniques. Clearly, this is a topic of great interest in all fields of mobile robotics, especially in manufacturing plants and Smart Factories, where a certain degree of decision autonomy of robotic systems and co-presence of robotic systems with humans is required. Specifically, this section will illustrate the problem that needs to be addressed in a general way, coupled with an accurate description and modeling of individual elements useful for numerical simulations, thus for training the mobile robot controller, and that is as accurate as possible in order to minimize the reality gap between the simulated and real environments. Particular emphasis in fact has been placed on modeling the moving crowd of pedestrians, particularly the interactions that occur between them, along with some social behaviors, but also the interaction that occurs between robots and humans. These interactions are modeled using a microscopic approach based on the introduction of contributions from appropriate attractive and repulsive forces. Next, the focus shifted to the architecture of the reinforcement learning environment, the topology of the neural network, and the algorithms that were used. Attention was then placed on the training and validation of the neural networks and a comparison of the different algorithms used was performed. Finally, the approach was then experimentally validated using an industrial mobile robot in a real-world scenario.

It is evident that certain types of interactions must be carefully resolved in this specific type of application. The mobile robot, as in the cases described in the previous sections, is subject to two types of interactions with the environment: that resulting from the contact of the wheels with the ground, and ensuring

locomotion and relative motion with it, and the interaction of the mobile robot with obstacles in the surroundings. Moreover, in this application, the mobile robot is also subject to interaction with humans present in the environments and classified under the group of human-robot interactions. It is evident that for this application, the interaction of the robot with the ground is not very relevant, while the interactions with obstacles and people assume a fundamental importance for the proper functioning of the robot in this scenario. However, among the latter two, the one that takes priority is human-robot interaction since it is required that the mobile robot should never be able to cause harm to a human being. Rather, it is preferable for it to crash into a static obstacle. Finally, it is important to note that all these interactions are modeled explicitly within the simulation environment, but it is the trained robot controller who, through its perception system, is able to detect and handle them appropriately in order to perform the task assigned to it.

Summarizing, the contributions of this section can be listed in:

- The creation of an extended social forces model, that aims to closely mimic the real case scenario, that enables the inclusion of more socially diverse human behavior, such as stopping, grouping, splitting, abrupt changes in environment direction, and so on, in addition to the introduction of the pedestrian-to-robot repelling force;
- The development of a functional and dimensionally effective CNN-based architecture to address the issue of mobile robots navigating moving crowds;
- The rigorous testing of DQL and A3C RL algorithms being used to solve the issue of mobile robots navigating moving crowds;
- The presentation of the parallel and asynchronous computing techniques utilized to accelerate training;
- The development of the ROS package, which can be used both in simulated and real-world settings, for robot control using the trained NN and which also includes mapping, visualization, localization, position estimation, and trajectory planning functionalities;
- Experimental validation of the RL trained controller on a commercially available mobile robot.

3.3.1 Introduction

It has posed a lot of attention and interest in robot autonomy and the potential use of robots in situations that entail interaction and collaborative work with humans from the initial days of mobile robotics. The ability of autonomous robots to move among humans and conduct activities safely and comfortably is a fundamental need for such applications. This is especially relevant in situations like Smart Factories. The employment of autonomous mobile robots (AMRs), in these environments, is pretty recent and is becoming more prevalent and appealing in industrial contexts [334, 335], typically in the form of “fleets” that are

managed by a fleet manager [86,329]. This has been also possible due to the ever increase in degree of autonomy these mobile robots are exhibiting in the past decades. Mobile robots employed in the context of Smart Factories can work independently or cooperatively with other machines, robotic systems, or human operators to complete tasks, and they can co-exist in the same environment. In the latter scenario, safety needs to be the major priority because the mobile robot can not hurt or provoke injuries to nearby humans. In addition, while working among people, the robot must be able to identify human intents and respond appropriately [336]. In the field of autonomous systems and autonomous navigation, it is well known the *freezing robot problem* [337]. This issue refers to the particular situation in which a robot gets stuck when it gets enough surrounded by people, which frequently occurs in densely populated areas. The problem of mobile robots navigating moving crowds has been studied by numerous authors in preceding decades due to the enormous potential of completely autonomous systems, which can be employed in a vastness of fields, such as the industrial one, commercial, health care, and so on. In 2013, Trautman et al. employed interactive Gaussian Processes to increase human-robot collaboration in dense crowd navigation situations [338]. To prevent collisions among a fleet of autonomous robots, Abdulov and Abramenkov in their research instead use a communication system [339]. This study is a logical extension of our earlier work [340], in which it had been trained an artificial neural network (ANN) utilizing the *NeuroEvolution of Augmenting Topologies* (NEAT) algorithm [341], therefore by employing an evolutionary approach to control a differential drive mobile robot.

The purpose of this study can be summarised as training a controller, to employ then on a mobile robot that will work in a realistic environment, by taking advantage of quick and inexpensive computer simulations. This type of environment is very well known in the literature research as Sim2Real [342] and has drawn a lot of attention recently. To ensure that the learned policy can be applied in the actual world, there must be a strong degree of congruence between both the simulated and real environments. In this instance, it's crucial to carefully select the approach utilized to mimic the crowd behavior in order to get a trustworthy environment.

An established method to mimic crowds in many applications, such as video games, is crowd simulation [343].

Crowd modeling and simulation may be done using a variety of methods, including force-based interactions, pedestrian flow, rule-based, psychology, and sociology-inspired methods, and others. The literature provides thorough overviews and discussions of crowd modeling and simulation methodologies [344,345]. This work relies on a social force model (SFM) which is based on the work presented by Helbing et al., as shown in [346,347]. They used a force-based model that is inspired by self-driven many-particle systems to model the behavior of crowds responding to panic situations; in addition, a similar method has been employed in other crowd models [348]. In order to estimate the variations in pedestrian speed, they specifically take into account the so-called *interaction forces*. These forces are used to reflect actual experimental data about humans walking, such as the propensity of pedestrians to maintain a velocity-dependent distance from each

other and from walls. Additionally, the authors also add a *repulsive interaction force* to represent the psychological finding that people on foot want to avoid one another. Last but not least, granular interactions that take place, particularly in tight crowds, are taken into account by including *body forces* as well as *sliding friction forces*. The software used to implement this crowd model was created by the same authors and assigned to it the name *PySocialForce*. Subsequently, it was then made publicly available for python as a cross-platform installable external module. Their program is implemented within this work, that in turn has been further enhanced.

Reinforcement learning (RL) approaches have been effectively leveraged in order to solve problems related to control and optimization aspects in a variety of domains, including video games, [349], chip placement [350], control of stratospheric air balloons [351], control of UAVs [110], and nuclear reactors [352]. This is a consequence of the growing resources that are available as well as computational power. The recent success of RL has been significantly aided by the usage of deep neural networks (DNNs), which can readily accommodate high dimensional inputs. In applications involving robots, the data from several sensors and actuators installed on these machines often make up the control system's input. As a result, it might be quite challenging to directly learn a strategy that would solve the required goal [353]. However, deep RL enabled "mapless" robot navigation by utilizing neural networks and perceptual data [354]. Previous research has attempted to train the robot's controller using the RL technique as well as simulation models to address navigation in congested situations [355,356]. Additionally, Regier et al. controlled a mobile robot using a NN with an input neural layer relying on the chunk notion as well as the SFM applied on simulating crowd motion [357].

There are further instances of RL being combined with PySocialForces, such as the work presented by Katyal et al. [355]. The strategy presented in this work, however, is different in a number of aspects. It has been expanded the SFM in order to account for more intricate social behaviors (agents can stop in the environment for a while, divide, group together, alter course, and so on). Proximal Policy Optimization (PPO), initially introduced by Schulman et al. [358], is used in the research from Katyal et al., however, within this work, it has been employed the following algorithms: the *Deep Q-Networks* (DQL), first presented by Mnih et al. [359], and the *Asynchronous Advantage Actor Critic* (A3C), first presented by Mnih et al. [360].

Finally, this section is structured as follows: in Sec. 3.3.2 the description of the problem considered in this investigation is provided. Will be first presented the modeling of the crowd using the SFM, then will be briefly described the kinematics of the mobile robot, then it will be described the modeling process of the simulation environment followed by the modeling of the robot's perception system, and finally will be presented the map chunk model used as a baseline for the additional task in this problem. In Sec. 3.3.3 is presented in detail the methodology aspects used within this investigation. These comprise first a brief description of the Markov Decision Process elements peculiar to this problem; then a description of the RL architecture used, hence the description of the algo-

rithms and the topology of the NN that has been used. In Sec. 3.3.4 are reported and described the findings obtained for the training process performed on the two RL algorithms, namely the DQL and A3C. In Sec. 3.3.5 are reported and described instead the findings that have been obtained for the testing process performed on both the selected RL algorithms. Finally, in Sec. 3.3.6 are reported the findings that have been obtained during the experimental validation on a real mobile robot of the proposed approach, and preceded by a description of the ROS-Gazebo pre-testing procedure performed for fine-tuning and check of the functionalities.

3.3.2 Problem statement

The problem statement is described within this paragraph. Particularly, first, it will be described the crowd model that had been chosen to describe the moving crowd; secondly, it will be reported the kinematics model of the hypothetical mobile robot along with the modeling of its perception system; subsequently, it will be reported the description of the simulated environment within the model; and last but not least the "map chunk" model that has been added in order to aid in speeding up the controller training process. Specifically, to the model have been added terms representing the following occurrences: pedestrians as well as groups stopping in the environment; individuals dynamically grouping with each other or with existing groups; single pedestrians abandoning their groups and also moving in other directions; or even groups dissolving into smaller groups; and finally, the ability for pedestrians as well as groups to meet up within the simulated environment.

Crowd modeling: Social Forces Model (SFM)

In this study, it has been used a simulation engine which is based upon the Social-Forces Model (SFM) in order to mimic dense crowds as well as their behavior. It has been specifically employed the *PySocialForce* [361], aforementioned python implementation package, which extends [362] and implements the Extended Social Forces model (ESFM). The original implementation of this model has been substantially expanded throughout this work to boost generality while taking into account common social behaviors that may be seen in real crowds during everyday activities.

The SFM is a microscopic approach that seeks to describe and simulate a moving crowd and has been extensively investigated in the past. The model postulates that Newton's second law, which describes dynamics as a sum of "forces" contributions, can be used to represent the motion of a single individual making up the crowd. The same research group created the ESFM, to which the aforementioned python library alludes. They have described each force component in their numerous papers and fine-tuned the model using experimental observations of actual crowds. These contributions are listed in the following, although full and in-depth discussions may be found in the authors' original manuscripts [361–364]. The ESFM presupposes that Newton's second law can adequately

capture the motion of a single pedestrian i, as shown in the following:

$$\ddot{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{f}_{\boldsymbol{t,i}} + \sum_{j=1}^{n} \boldsymbol{f}_{\boldsymbol{i,j}} + \boldsymbol{f}_{\boldsymbol{w,i}} + \boldsymbol{f}_{\boldsymbol{g,i}} \tag{3.37}$$

where $\boldsymbol{f}_{\boldsymbol{t,i}}$, defined as a specific external force, characterizes the contribution to the mobility of the i-th pedestrian in order to reach its assigned target destination; $\boldsymbol{f}_{\boldsymbol{w,i}}$ denotes the repulsive force component resulting from a static obstacle w existing in the simulated environment, which helps the individual to stay away and avoid collisions with obstacles; $\boldsymbol{f}_{\boldsymbol{i,j}}$ represents the repulsive force contribution with the pedestrian j, which has the effect of single pedestrians not colliding with each other; and finally $\boldsymbol{f}_{\boldsymbol{g,i}}$ is defined as the grouping force contribution, which has the effect to maintain the formation of a given group. The result of the total of these forces translates into the final motion of the i-th pedestrian. In this context, it is referred to as "specific force" the external force scaled by the i-th pedestrian mass m_i, i.e. the forces contributions in Eq. (3.37) formally represent acceleration terms.

The target contribution term $\boldsymbol{f}_{\boldsymbol{t,i}}$ is suggested to be characterized by the relation below as claimed by Helbing et al. in [362]:

$$\boldsymbol{f}_{\boldsymbol{t,i}} = \frac{v_i^t \hat{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{t}} - \boldsymbol{v}_{\boldsymbol{i}}(t)}{\tau} \tag{3.38}$$

where v_i^t is the scalar i-th individual's target speed, $\hat{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{t}}$ is the desired direction's normalised vector, $v_i(t)$ is the individual's current speed vector at time t, and finally τ is a tunable relaxing factor.

The individual repulsive interaction force contribution between the generic pedestrians i and j, according to Moussaïd et al. in [364], may be expressed as,

$$\boldsymbol{f}_{\boldsymbol{i,j}} = -Ae^{-d/B}\left[e^{-(n'B\theta)^2}\boldsymbol{t} + e^{-(nB\theta)^2}\boldsymbol{n}\right] \tag{3.39}$$

where $\boldsymbol{t}$ is the direction of the interaction, i.e., the versor pointing from pedestrian i towards pedestrian j; $\boldsymbol{n}$ is defined as the normal versor to $\boldsymbol{t}$ and oriented to the left, and d represents the separation between the two pedestrians. A, B, n, and n' are tunable model parameters; θ is the angle between $\boldsymbol{t}$ and the versor that points from i towards j.

According to Johansson et al. in [363], the repulsive contribution force to the i-th individual caused by closeness to walls and obstacles in the environment may be represented as the following decaying exponential relationship:

$$||\boldsymbol{f}_{\boldsymbol{w,i}}|| = ae^{-d_w/b} \tag{3.40}$$

where a and b are model parameters, and d_w is defined as the normal distance between the i-th pedestrian and the obstacle. Moreover, this force has a direction coinciding with the normal distance and acts from the obstacle to the pedestrian.

Finally, Moussaïd et al. proposed the grouping contributions in their research [361]. These contributions have been described as consisting of three terms as follows,

$$\boldsymbol{f}_{g,i} = \boldsymbol{f}^{\boldsymbol{g}}_{\boldsymbol{g},\boldsymbol{i}} + \boldsymbol{f}^{\boldsymbol{a}}_{\boldsymbol{g},\boldsymbol{i}} + \boldsymbol{f}^{\boldsymbol{r}}_{\boldsymbol{g},\boldsymbol{i}} \tag{3.41}$$

where $\boldsymbol{f}^{\boldsymbol{a}}_{\boldsymbol{g},\boldsymbol{i}}$ is the group attraction term, which is in charge of attracting individuals who belong to the exact same group; $\boldsymbol{f}^{\boldsymbol{r}}_{\boldsymbol{g},\boldsymbol{i}}$ is defined as the group repulsive term, which is in charge of preventing group members from colliding and overlapping, and finally $\boldsymbol{f}^{\boldsymbol{g}}_{\boldsymbol{g},\boldsymbol{i}}$ is defined as the gazing term, which is in charge of adjusting the i-th pedestrian pose in order to minimize its head rotation. For a complete definition of the single terms the reader is referred to the following literature [361].

It must be pointed out that simulated pedestrians have been shown to demonstrate lane-following behavior while attempting to avoid static obstacles present in the environment. In addition, some of them have been observed to become trapped in a local minimum while in close proximity to objects with complicated shapes. In this study, individuals have been provided with an ad-hoc controller to assist them in more efficiently avoiding static impediments. The last difference between the implementation presented within this work of the individual dynamics and the original ESFM formulation is that this approach takes into consideration the fact that the solitary pedestrian is somehow aware of the presence of the robot. This has been accomplished via a suitable weak repulsive force $\boldsymbol{f}_{\boldsymbol{i},\boldsymbol{r}}$ that only operates close to the mobile robot, that is, whenever the separation falls below the tunable threshold parameter d_R. The robot-to-pedestrian repulsive force contribution can then be added to the formulation of the pedestrian dynamics in Eq. (3.37) in order to account for this addition. However, it must be clarified that during the training and validation procedures, it has been elected to choose to leave this force component disabled, which means that now the pedestrians are unaware of the robot's movement in the environment. This component was purposefully turned off because it serves as a cautionary case in which the mobile robot must learn in a more difficult situation. The crowd doesn't react to the robot because they perceive it to be a ghost.

Mobile robot kinematics

A Wheeled Mobile Robot (WMR), whose kinematics is of type differential, which means that it possesses two independent driving wheels and a common axis of rotation, has been taken into consideration within this study as a case study. Due to its well-known kinematics and due to the non-holonomic constraints, this WMR can not move laterally in this configuration but may drive straight, steer, and spin in situ.

Two reference frames have been introduced—a *local reference frame* fixed with the WMR $(\boldsymbol{C}, \hat{\imath}, \hat{\jmath})$ and an *inertial reference frame* $(\boldsymbol{O}, \hat{e}_x, \hat{e}_y)$—to characterize its motion. Moreover, the robot heading is represented by the rotation between the two reference frames. This is demonstrated in Figure 3.17.

The differential kinematics model formulation, for a differential drive mobile robot can be expressed in the robot's local reference frame under the conditions of pure rolling and no lateral slip. Referring to Figure 3.17, it follows that,

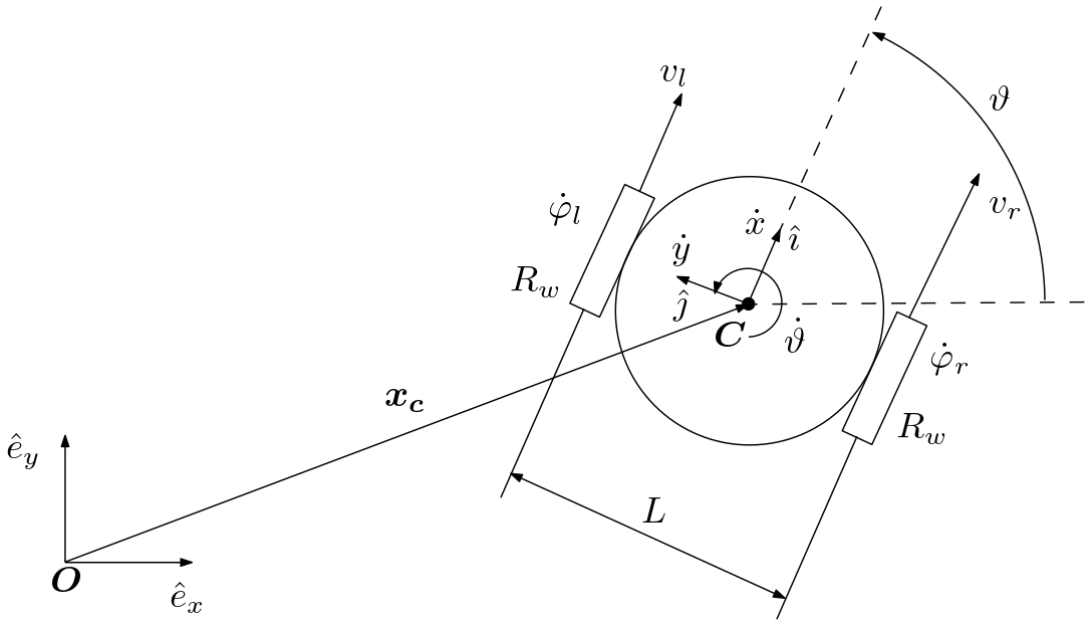


Figure 3.17: Kinematic model of a differential drive mobile robot, with focus on the relation between the robot's frame and the inertial frame [365].

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} \frac{R_w}{2}\left(\dot{\varphi}_l + \dot{\varphi}_r\right) \\ 0 \\ \frac{R_w}{L}\left(\dot{\varphi}_r - \dot{\varphi}_l\right) \end{bmatrix} \tag{3.42}$$

where L refers to the wheelbase of the mobile robot, $\dot{\varphi}_l$ and $\dot{\varphi}_r$ are the left and right wheels' respective angular speeds, and R_w represent the radius of the wheels. Additionally, as indicated in Figure 3.17, $\dot{x}$, $\dot{y}$ and $\dot{\vartheta}$ are defined as the WMR's linear and angular speeds given in its local coordinate system. By using a rotational transformation, the same speeds and, consequently, the kinematic model itself, may be stated in the inertial frame as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{I}} = \boldsymbol{R}^{\boldsymbol{T}}\dot{\boldsymbol{x}} \tag{3.43}$$

where,

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.44}$$

where $\boldsymbol{R}$ is defined as the bi-dimensional rotation matrix between the robot and the inertial reference frames.

It must be pointed out that Eq. (3.43) serves as the starting point for the localization of the WMR and it also describes its differential kinematics expressed in the inertial frame. In fact, a process has been put into place that simulates the estimation of the mobile robot's position, by using an odometry subsystem from its wheels encoder readings. More precisely, the WMR can localize itself with respect to the inertial frame as follows,

$$\boldsymbol{x}(t) = \int_{t_0}^{t} \boldsymbol{R}(t)^T \dot{\boldsymbol{x}} dt + \boldsymbol{x}(t_0) \tag{3.45}$$

which in the discretized case, and by using a simple fixed-step explicit Euler integration scheme, it becomes:

$$\boldsymbol{x}(t_k) = \sum_{i=0}^{k} \boldsymbol{R}(t_i)^T \dot{\boldsymbol{x}}(t_i) \Delta t + \boldsymbol{x}(t_0) \tag{3.46}$$

It is important to point out that accuracy in the integration process can be raised by utilizing a better integration scheme. Within this study, it has been elected to use a fixed-size time-step trapezoidal integration scheme.

The LiDar distance sensor installed on the WMR only scans the area in front of the vehicle, it follows that the modeled WMR has been elected to be constrained only to positive values of linear velocity as a precautionary measure. Additionally, the WMR velocity is restricted to some maximum values in order to upperly limit the permitted robot velocities and therefore maintain it near the maximum. The following equations provide a summary of these constraints:

$$\begin{cases} \dot{x} = v \in [0, v_{max}] \\ \dot{y} = 0 \\ \dot{\vartheta} = \omega \in [-\omega_{max}, \omega_{max}] \end{cases} \tag{3.47}$$

The Table 3.2 provides a summary of the WMR, 2D-LiDAR, and odometry subsystem parameters.

From a high-level control perspective, the WMR gets set-points for its linear and angular velocities $[v^*, \omega^*]$, which are evaluated by utilizing Eq. (3.42). Moreover, these set-points are being changed on every time-step while taking into consideration the differential input. This results in a limited set of inputs from which the NN can choose. The aforementioned differential input is defined as:

$$u = [\Delta v, \Delta \omega] \tag{3.48}$$

Consequently, after expressing Eq. (3.48) by utilizing the saturation functions sat_v for the linear speed and sat_ω for the angular speed, it follows that:

$$v^*(k) = sat_v(v^*(k-1) + \Delta v(k)) \tag{3.49}$$

$$\omega^*(k) = sat_\omega(\omega^*(k-1) + \Delta \omega(k)) \tag{3.50}$$

Environment modeling

The environment that has been used for training the NNs, by exploiting simulations, is described as a bi-dimensional grid with dimensions of width W and height H, in which are present static obstacles, as well as moving pedestrians. The environment's reference frame, considered for this study, coincides with the inertial frame. In order to expose the WMR to a wide range of conditions and also

to generalize the issue, all static objects are created with random locations and geometries at the episode's initialization. Instead, pedestrians may enter and leave the area at the borders. The WMR, on the other hand, is not provided with any information about the environment because it acquires local knowledge through its sensing system, which is covered in more detail in the following paragraph. From Figure 3.18, it can be seen an example of a training environment where the WMR must travel from a randomly generated starting point, P, to a randomly generated target location, T, without colliding with the crowd in motion, which is represented by the points p_i, which stand for individual pedestrians, and the group of pedestrians, g_i. Since the target's location and the robot's position are both determined at random, it has been chosen to place a criterion on the initial minimum distance between the two in order to guarantee that all trajectories are similar in length and degree of difficulty. This condition is therefore expressed as follows:

$$dist(P, T) > d_{min} \tag{3.51}$$

The decision to generate the robot's initial position and target location in a random way has been taken to lessen the probability that the robot may find alternative and simple paths to reach its destination (e.g. move close to the edges of the environment). As shown in Figure 3.19, it has been set a safety radius R_R for the robot and a safety radius R_P for individuals in order to enforce a safe distance between the WMR as well as the surrounding environment and account for the physical characteristics of both. Instead, as done in Sec. 3.2, static obstacles have been enlarged to account for a safety turning radius that is required for obstacle avoidance maneuvers. In fact, the grid cells that these entities occupy in the grid-represented world are flagged as occupied.

Table 3.2: Summary of the parameters describing the whole environment simulation, including the robot, pedestrians, and range sensor [366].

Parameter	Value	Parameter	Value
R_w	0.1m	R_p	0.3m
L	0.5m	R_R	0.6m
v_{max}	$1\ \mathrm{m\,s^{-1}}$	ω_{max}	$1\mathrm{rad\,s^{-1}}$
r_{min}	0.3m	r_{max}	10m
$\Delta\alpha$	$[-\pi/2, \pi/2]$	n_{rays}	135
$W \times H$	$20\,\mathrm{m} \times 20\,\mathrm{m}$	d_{min}	32m

Robot perception

The WMR is thought to be outfitted with a 2D LiDAR laser scanner, that provides it with perception and collects data from its surroundings. Object detection is achieved by employing this device, which in the simulated environment uses a *ray-casting* technique. The series of the most recent k LiDAR measurements, where k is an adjustable value, are provided as input to the controller's policy in

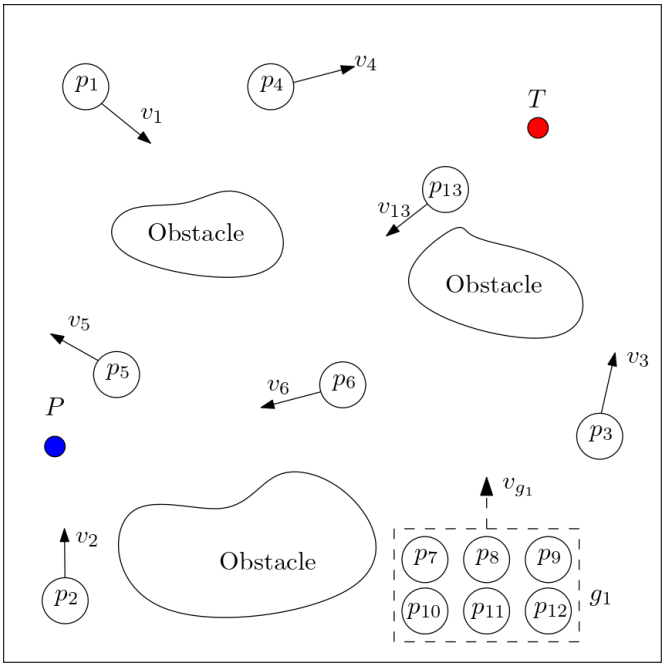


Figure 3.18: Schematic representation of the simulated environment: the robot spawning location and its target destination are highlighted in blue and red respectively; three static obstacles are represented. Each pedestrian p_i is indicated as a circle having its own speed v_i, while the group is represented with an enveloping dashed rectangle and depicted with g_i [366].

order to enable the WMR to respond to dynamic and moving obstacles such as pedestrians.

In the previous work from our team [340], it has been demonstrated that the ray-casting algorithm can be used successfully in detecting pedestrians. It has been demonstrated also that starting from point clouds, obtained through sensor readings, and through the apposite post-processing process it is possible to reduce the reality gap between a ray-casting-based approach and real sensors, and also between an idealized person and a real person. Finally, it is possible also to reconstruct the idealization of pedestrians and distinguish single pedestrians starting from sensor point clouds.

The maximum sensor range, r_{max}, and the scanning angle, $\Delta\alpha$, determine the polar area within which the rays of the simulated lidar sensor installed atop the WMR span. By defining the scan resolution as s_r, it is apparent that:

$$s_r = \frac{\Delta\alpha}{n_{rays}} \tag{3.52}$$

where n_{rays} is defined as the number of rays being shot by the sensor. This parameter is responsible to controls the density of the rays inside the scanning area. Specifically, Figure 3.19 shows, from a graphical and easier way, the above-described setup as follows: a range r_i, defined as the distance between the sensor's reference frame origin $\boldsymbol{C}$ and the point of intersection, is returned when a ray encounters an obstacle p_i; otherwise, the maximum range r_{max} associated with

the sensor is returned. Since the angle α_i is implicitly known (in reality, it is an arbitrary value defined), the information regarding all intersection points in a polar representation (r_i, α_i) is easily obtainable. As will be seen in the later paragraphs, if the rays are ordered, for the scope of training the NNs, the information about the angles is redundant, thus only ordered ranges are used.

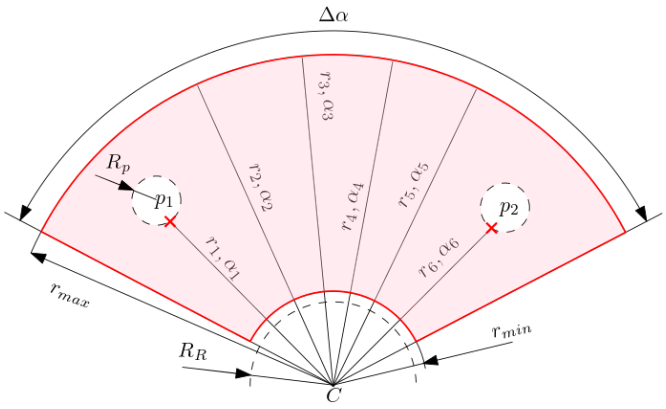


Figure 3.19: Working principle of the 2D LiDar range sensor. The figure shows a general scanning area, positioned ahead of the WMR, defined by $(r_{min}, r_{max}, \Delta\alpha, n_{rays})$, in which two pedestrians are detected [366].

A minimum scanning range r_{min}, that is only a little bit larger than the robot's dimensions, is introduced in order to eliminate erroneous readings caused by the robot geometry. Additionally, it has been elected to not model an ideal 2D LiDAR within the simulated environment; rather, it has assumed that every ray (r_i, α_i) is susceptible to erroneous both positive and negative measurements with a probability of p_{fp}, p_{fn}, respectively.

The scanner on the actual WMR that has been modeled within the simulated environment only scans the robot's forward direction. This makes the investigated problem even more complex and makes it tougher for the mobile robot to maneuver through the moving crowd. Furthermore, due to geometric restrictions as well as the placement of the LiDAR, the scanner is unable to obtain a complete polar view of the area surrounding the WRM (see Figure 3.32). Finally, the characteristics and specifications for the sensor and the robot environment utilized for simulations are listed in Table 3.2.

Map chunk model

The model for the robot's perception (also known as the "chunk model") that has been implemented in the RL framework, as explained and shown in the previous paragraph, is shown in Figure 3.20. Specifically, both stationary obstacles as well as individuals are represented in the scenario as is shown. The robot's scanning region, or the area covered by the LiDAR device around it, is divided into a number of equal n_q sectors, each of which contains details about the nearest object. This method has been created with the aim to minimize the dimensionality of significant data related to the condition of the environment and potential robot collisions.

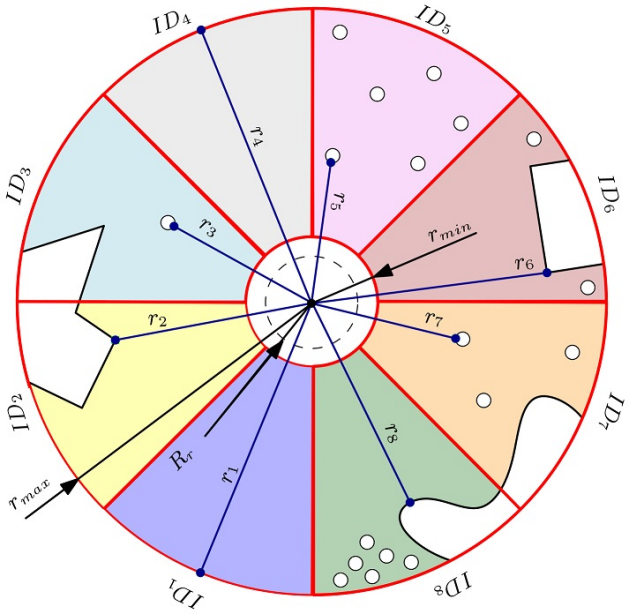


Figure 3.20: Schematic view of the chunking principle of the field of view of the robot. In this example the area is split into eight sections: for each section, the distance from the closest obstacle/pedestrian is taken [366].

With the aforementioned chunking technique, it is possible to get two perceptual outputs, denoted by the vectors d^p and d^m, respectively, if just individuals are taken into account or the closest of all map obstacles, i.e. fixed obstacles, environment edges, as well as pedestrians. Specifically:

$$d_i^m, \quad i \in [1, \ldots, n_q] \tag{3.53}$$

$$d_i^p, \quad i \in [1, \ldots, n_q] \tag{3.54}$$

where $d_i^m \leq d_i^p$, $\forall i$.

The chunking method can therefore be used in two ways to accelerate the controller's training process even if the perception system can only scan 180 degrees directly in front of the robot at a time. In fact, for this particular issue, it has been designed two auxiliary tasks that have separate final layers of the NN but they are sharing its same weights. These tasks can be summarised in:

- The first auxiliary task entails using previous observations to estimate the location of all nearby obstacles (here, d^m is utilized);
- The second task alternatively seeks to optimize a policy that maximizes one-step reward punishing states under which the WMR is surrounded by pedestrians coming from various directions (d_p is chosen because being close to a static obstacle does not always indicate a threat of collision).

3.3.3 Methodology

Reinforcement learning techniques can be used to tackle the navigation problem in a dynamic and crowded environment because the problem can be considered as

a Markov Decision Process (MDP) [367], in which an agent observes states s and hence takes actions a. Reinforcement learning can be used in this type of scenario to identify a policy, namely a mapping from states to actions, which controls the agent and that maximizes a certain objective described by the *reward function* associated with the MDP.

Using probabilistic terminology, an MDP describes a transitions system that is defined by a tuple (S, A, P_a, R_a), where S is referred to as the state space, A is referred to as the action space, $P(s'|s, a)$ is referred to as the probability of transitioning to state s' if an action a is selected at state s, and $R(s, a, s')$ is referred to as the reward associated with the transition from s to s', which in turn is denoted with $s \xrightarrow{a} s'$. Moreover, the internal state of the WMR, the static obstacle layout, as well as the pedestrian dynamics serve as the state representation s inside this crowd navigation issue. The transition probability $P(\cdot)$ is determined by the combined WMR's dynamics and the individuals, whereas the actions taken by the agent are the potential signals that the controller may provide to the robot. Because of the randomized behavior of the individuals and the dependence on the number of agents involved, it is not physically possible to deduce the probabilistic model $P(\cdot)$ within such a complicated MDP model. Therefore, it has been used the RL framework and, moreover, it has been taken advantage of DNNs' flexibility to learn the best policy π.

The crowd navigation MDP, the RL algorithms being selected and employed, the NN's architecture utilized for the robot controller, as well as information on the implementation of the paralleled training environment which has been designed are all covered in the following:

Elements of the Markov Decision Process

When using RL to solve MDPs, the problem's viability depends heavily on the description of the MDP's single parts. In the following paragraphs will be described how these elements of the MDP have been modeled within this investigation.

- **State Space**: The state space S, or the space of the potential inputs, should be informative and contain the robot's relative position, orientation, and speed in relation to the commanded destination, as well as static objects, individuals, and other objects that might be present in the environment. This will enable the WMR to travel in a heavily populated area. Additionally, the state that the controller observes must have sufficient details to enable the controller to properly predict how individuals will move. This investigation task has two primary objectives: (i) getting the WMR to the specified target, and (ii) avoiding collisions with objects and individuals. The information needed to execute each of these duties can be determined by looking at these two objectives independently, even if they are not entirely independent from one another. The policy $\pi(s)$ would solely require the information regarding the robot's to reach the commanded target state if it is assumed there are no obstructions or individuals nearby. However, this is not the case, it follows that are needed more information. More precisely are needed more information about the environment, which in turn

are provided by the range sensor in this case. It is possible to express the internal state of the robot as:

$$s_{rbt} = [x_T, \vartheta_T, v, \omega] \tag{3.55}$$

where the orientation and relative position of the target point with respect to the robot are denoted by x_T and θ_T, respectively, in this instance.

As stated above, for the problem of WMR's navigating moving crowds, the state space S would require also additional information about the environment, which are not internal to the robot. More in detail, the robot policy must be aware of the dynamics of each element that is present, both static and moving, in the environment in order to prevent collisions. Particularly, it is impossible to precisely determine each object's or individual's position and speed based on the limited amount of data that the LiDAR range sensor provides. Nevertheless, by supplying the agent also previous observations of the perception system as well as the evolution of the WMR's internal state, it is feasible to extrapolate those quantities. As a consequence, it is, therefore, possible to define the environment observation state as:

$$s_{env} = \{s_1, \ldots, s_m\} \tag{3.56}$$

with,

$$s_i = (l_{1,i}, \ldots, l_{p,i}, v_i, \omega_i) \tag{3.57}$$

where m refers to the number of the past observations, v_i and ω_i are the WMR's linear and angular speed at time i, while $l_{j,i}$ is referred to as the j-th range measure of the lidar sensor obtained at time i, where $p = n_{rays}$ and $j = 1, \ldots, p$. In conclusion, the pair that defines every element s of the State Space S is defined as follows,

$$s = (s_{env}, s_{rbt}) \tag{3.58}$$

- **Action Space**: It has been elected to discretize the robot's actions to maintain the complexity of the learning problem reasonable. Specifically, it has been chosen nine combinations of three values for each action, specifically for the linear and angular speed variations, that the differential drive WMR may produce. Consequently, the action a is therefore expressed by the following pair,

$$a = (\Delta v, \Delta\omega) \tag{3.59}$$

where

$$\Delta v \in \{-\Delta v_{max}, 0, \Delta v_{max}\} \tag{3.60}$$

$$\Delta\omega \in \{-\Delta\omega_{max}, 0, -\Delta\omega_{max}\} \tag{3.61}$$

- **Reward Function**: A proper structure of the reward function is crucial in determining the complexity of the learning task in RL approaches. For instance, a straightforward reward definition may merely offer either negative

or positive feedback in the event of a collision or when the commanded destination is reached, respectively. However, a signal this sparse restricts the agent's capacity to learn, which translates into the fact that finding the best policy becomes considerably more challenging. Additionally, a very well-structured reward function can significantly hasten the convergence of the agent's policy. The simulated environment it has been taken into consid-

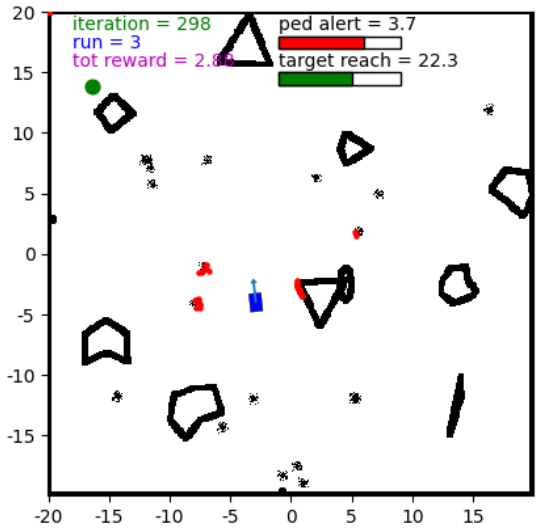


Figure 3.21: Learning environment: the blue rectangle represents the robot and the arrow its current heading; the red dots the LiDAR scanner readings; the green circle represent the target location the robot must reach; finally, in black are represented the obstacles. Obstacles can be either static or dynamic (black dots represent pedestrians) [366].

eration within this study is constituted of a square-shaped area, and while pedestrians can freely enter and leave the environment through its perimeter, the robot is not allowed to do so, in fact, it has been assumed that it sees it as an obstacle. Static obstacles are randomly positioned inside the scene and have a variety of random polygonal geometries (see Figure 3.21). Additionally, the generic training episode starts with the creation of a fresh map describing the environment; subsequently, random target and robot initial coordinates, that fulfill the minimal initial distance, are generated. Furthermore, a single episode run, also called trajectory from now on, has been elected to be considered fully successful only if the robot manages to reach the commanded target location without colliding with either pedestrians or static obstacles. For this reason, few termination criteria have been introduced in the simulation scenario.

In the MDP framework, it has been designated a reward function R which, taking into account the aforementioned factors, gives "feedback" to the agent after every transition $s \xrightarrow{a} s'$. Specifically, it has been taken into account three possible scenarios:

– *Success*: When the robot successfully reaches the destination, the

episode ends in state s', and the agent is awarded with a positive reward, defined as $R = +K$;

- *Collision or timeout*: The episode finishes in the state s' because of a collision or simply because the simulation time has expired, where the latter means that the maximum allowed number of simulation iterations have been exceeded. If this condition occurs, then the agent is penalized with a negative reward defined as $R = -K(0.75 + FD)$.
- *Keep progressing*: If none of the above has occurred, then the robot can continue to advance and earn a reward, which in turn is defined as $R = +k(DB - SP - PPM)$.

where FD is the robot's distance to the goal acquired at the conclusion of the episode, normalized in relation to the greatest distance feasible in the environment space, i.e. the square diagonal. On the other hand K, k are defined as the final and intermediate constant rewards. The following definitions apply to the remaining terms defined in the third scenario, which serve as intermediate bonus and punishment components:

- *Direction bonus (DB)*: is defined as $DB = 1$ if, following the action a, the horizontal separation between the robot and the target has decreased, otherwise it is set to $DB = 0$;
- *Saturation Penalty (SP)*: if the action a that has been chosen by the actor has caused the actuators to fall in saturation condition, then *SP = 1*, otherwise *SP = 0*;
- *Pedestrians proximity penalty (PPM)*: Is a penalty that depends on how the WMR is close to an obstacle and is defined as follows,

$$PPM = c_P \sum_{i=1}^{n_q} (1 - d_i^p)^3 \tag{3.62}$$

where c_P is a constant and d_i^p is the normalized separation to the nearest pedestrian inside the i-th chunking sector, as specified in Eq. (3.54). Rather than d_i^m it has been considered the term d_i^p since it just takes individuals into account. The justification for this is that a trajectory that runs near a static obstacle, in order to avoid hitting people, shouldn't be punished. Indeed, it is always better to crash into a wall, rather than into a pedestrian. Lastly, the cubic exponent makes sure that individuals who are more than nearly 1/3 of a LiDAR range away from the sensor do not have an adverse effect on the intermediate rewards.

Reinforcement Learning architecture

Throughout this section, it is illustrated how the selected NNs have been trained by using two RL algorithms: respectively the parallel Deep Q-Learning (DQL) as well as the asynchronous Advantage Actor Critic (A3C). The NNs code has

been written using the PyTorch package for Python, even if this investigation is not intended to provide a full demonstration of the implementation details. The parallelization detailed in the next paragraphs was achieved using Ray [368], a general-purpose distributed application building API with a focus on real-time (RL) applications, which enables making just minor code changes through Python decorators to make it parallel.

- **Deep Q-Learning (DQL)**: The first RL technique used on deep NNs was Deep Q-Learning (DQL) [359]. It is based on Q-Learning, i.e. a traditional reinforcement learning technique, which is used to calculate the *State-Action value function* denoted with $Q^{\pi}(s,a)$, often known as the Q-function, of the best policy π^*. A specific π policy's Q-function will attempt on estimating the average discounted return, also known as the total of all future rewards, that may be obtained by using the policy π to select actions after starting in state s and then executing action a. The value function $Q^*(s,a)$ resulting from π^* fulfills the *Bellman optimality equation* and returns the optimal action to take, and is defined as follows,

$$Q^*(s,a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s',a')\right] \tag{3.63}$$

where the expectation is calculated in relation to the rewards' probability distribution r as well as the dynamics of the environment, and γ is defined as the discount factor for future rewards. When hypothetically $Q^*(s,a)$ is available, then the best policy π^* chooses action a as:

$$a = \underset{a'}{\operatorname{argmax}}\, Q(s,a') \tag{3.64}$$

The eminent studies of Mnih et al. have demonstrated that the recurrent form of Eq. (3.63) converges to Q^* [359], which is in turn reported below:

$$Q_{i+1}(s,a) \leftarrow \mathbb{E}\left[r + \gamma \max_{a'} Q_i(s',a')\right] \tag{3.65}$$

As a consequence, by iteratively refining the estimated Q-function, the best policy may be learned. The convergence in Eq. (3.65) is obtained if the Q-function is estimated by using a NN $Q(s,a;\theta) \approx Q^*(s,a)$, and this is done by resolving a regression problem. Specifically, observed transitions (s,a,r,s') are utilized to generate a goal value by utilizing the approximate Q-function to anticipate future rewards. This technique is known also as *bootstrapping*. This step serves as an ex-post evaluation of action a's value and yields the following loss function:

$$L(\theta_i) = \mathbb{E}\left[(y_i - Q(s,a;\theta_i))^2\right] \tag{3.66}$$

where the expectation must be calculated across the MDP variables s, a, r, and s', and the Temporal Difference (TD) target y_i is specified as follows,

$$y_i = r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) \tag{3.67}$$

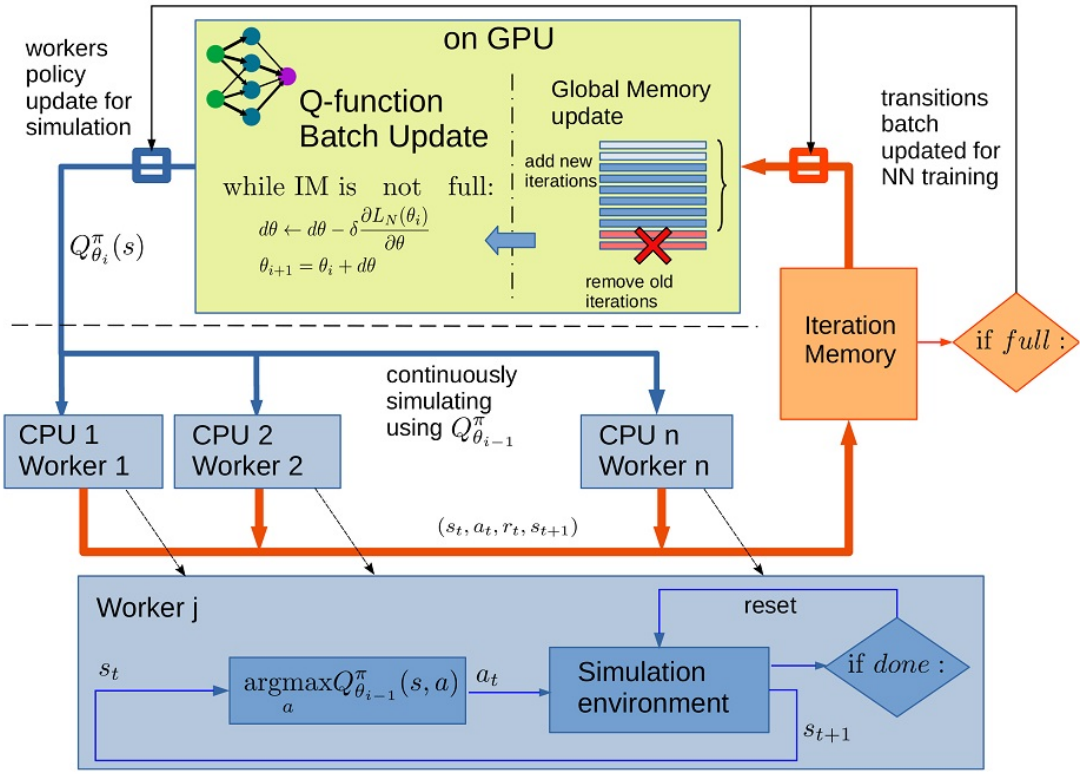


Figure 3.22: Schematic representation of the multi-node asynchronous DQL algorithm. The upper part of the diagram (NN update) and the lower (agents simulations) are performed in parallel on different nodes, at the end of each iteration data is shared (NN coefficients from the updater node and transitions memory from the simulating nodes) [366].

The DQL technique makes use of the so-called "experience replay" [369] for estimating the expectation in Eq. (3.66), therefore allowing for the minimization of the loss $L(\theta)$ through well-known Stochastic Gradient Descent (SGD) method. This method calls for the storage of recorded transitions $(s_t, a_t \rightarrow s_{t+1})$ in a circular buffer (memory), which is afterward utilized to train the Q-function using mini-batch SGD. This strategy offers two key benefits: the first is that each transition is utilized for numerous updates; the second instead is that employing uncorrelated transitions in a batch reduces variance.

In DQL, exploitation and exploration are balanced during training by utilizing an *epsilon-greedy* strategy [370] to select an action a while interacting with the environment. Hence:

$$a_t = \begin{cases} a = & \text{argmax}_{a'} Q(s,a') & , & \text{w.p. } \epsilon_t \\ a \sim & \mathcal{U}\{a_1, a_n\} & , & \text{w.p. } 1-\epsilon_t \end{cases} \tag{3.68}$$

Additionally, in order to reduce the probability of exploration in favor of exploitation, the ϵ_t is scaled down by a factor c_e, which will be referred to as *epsilon decay* from now on, at each iteration. This keeps on going until

a minimum value for ϵ_{min} is reached. It follows that:

$$\epsilon_t = \max(c_\epsilon \cdot \epsilon_{t-1}, \epsilon_{\min}) \tag{3.69}$$

Within this study, it has been implemented an asynchronous implementation of the DQL algorithm, where a schematic representation of it is shown in Figure 3.22. The training process is composed of training iterations, which in turn are defined by the number of transitions observed during a simulation with a specific Q-function version. Each iteration i stores the transitions in an *Iteration Memory* (IM) and exports the reference Q-function $Q^{\pi}_{\theta_{i-1}}$ needed to perform the simulations to numerous simulation environments instances deployed on various worker nodes. The main thread updates the weights of the NN on the GPU with batches taken from the "Global Memory" (GM) during the period needed to fill the IM, yielding the new Q-function $Q^{\pi}_{\theta_i}$. The IM replaces the GM's oldest transitions such that the following iteration can proceed.

This implementation has several benefits, listed in the following: (i) all computational resources are utilized at all times; (ii) fixed-sized iterations are an obvious choice for evaluating and storing training progress; (iii) memory relocation from the CPU to the GPU can take place only between iterations; and (iv) when using a cluster, the local memory capacity at every worker node can be adjusted to the resources at each node. In the following paragraphs, deployment-related aspects are summarized.

- **Asynchronous Advantage Actor Critic (A3C)**: The exact discrete action space which has been illustrated in the previous paragraphs has been leveraged for the case of the Asynchronous Advantage Actor Critic (A3C) algorithm [360]. This approach differs from the previous and it's called to be a Policy Gradient (PG) method, in which the policy is directly optimized by calculating gradients that update its weights. This objective function, which needs to be maximized, is defined as follows,

$$J^{\pi}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta, P}\left[G(\tau)\right] \tag{3.70}$$

that represents the average return that the policy has achieved, and it may be estimated by taking environmental trajectory samples while applying the policy π_θ:

$$J^{\pi}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i - 1} \gamma^t r(s_{i,t}, a_{i,t}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i - 1} G_{i,t} \tag{3.71}$$

where $G_{i,t}$ is referred to as the compounded total of all rewards received along a specific trajectory, starting at any intermediate point and until the end.

The REINFORCE algorithm, also known as Vanilla PG, is the basis of the A3C algorithm, where the former maximizes Eq. (3.71) using SGD. The primary objective of the improvements, coming from the introduction of

A3C, is to lower the large variance that distinguishes Monte Carlo sampling. Considering the advantage, denoted with $A(s_t, a_t)$, of doing a given action in comparison to a baseline, rather than $G_{i,t}$ calculated from the raw rewards (whose signs may be arbitrary), represent one approach to decrease variation. The actor-critic algorithms put this idea into practice in a way where the baseline is set by a *critic* function that roughly estimates a state-value function $V_\phi^\pi(s)$ for a policy π, i.e. the *actor*. Furthermore, causality makes it possible to evaluate the advantage at the instant t without taking into account past rewards. The advantage $A(s_t, a_t)$, by using Actor-Critic with "reward to go", therefore becomes:

$$A(s_t, a_t) = r - V_\phi(s_t) = \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r(s_{t'}, a_{t'}) \right) - V_\phi(s_t) \tag{3.72}$$

it is therefore possible to define a loss function, which needs to be minimized, as follows,

$$L_{\text{A3C}} = -J^\pi(\theta) + J^\phi(\theta_c) - \beta H^\pi(\theta) \tag{3.73}$$

where $J^\phi(\theta_c)$ is referred to as the *advantage loss*, and is defined as follows,

$$J^\phi(\theta_c) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} A(s_{i,t}, a_{i,t})^2 \tag{3.74}$$

and $H^\pi(\theta)$ is referred to as the entropy component that has been included in order to prevent an early convergence to a local minimum, and is in turn defined as follows,

$$H^\pi(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \left(-\sum_a \pi(s_i, a) \log \pi(s_i, a) \right) \tag{3.75}$$

An approximation of the gradient formula that maximizes Eq. (3.71) is as follows:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \nabla_\theta \log \pi_\theta(a_{i,t}, s_{i,t}) A(s_{i,t}, a_{i,t}) \tag{3.76}$$

The selected implementation is illustrated schematically in 3.23, where the A3C setup is obtained by executing threads $i = 1, \ldots, N$ in parallel, aggregating the gradients which have been computed on various worker nodes, and carrying out the "step" of the optimization locally inside the main thread. Moreover, the software retains the "iteration" structure which has been specified in the previous section for the DQL case, essentially terminating the iteration whenever the count of the simulated steps reaches the size of the "Iteration Memory" IM. It must be pointed out that this has been done in order to compare the performances of the various RL approaches.

- **Neural Network Topology**: In here it will be briefly described the topology of the NN, i.e. the structure, that has been chosen to use and train with the aforementioned RL algorithms. This NN is made up of essentially three parts:

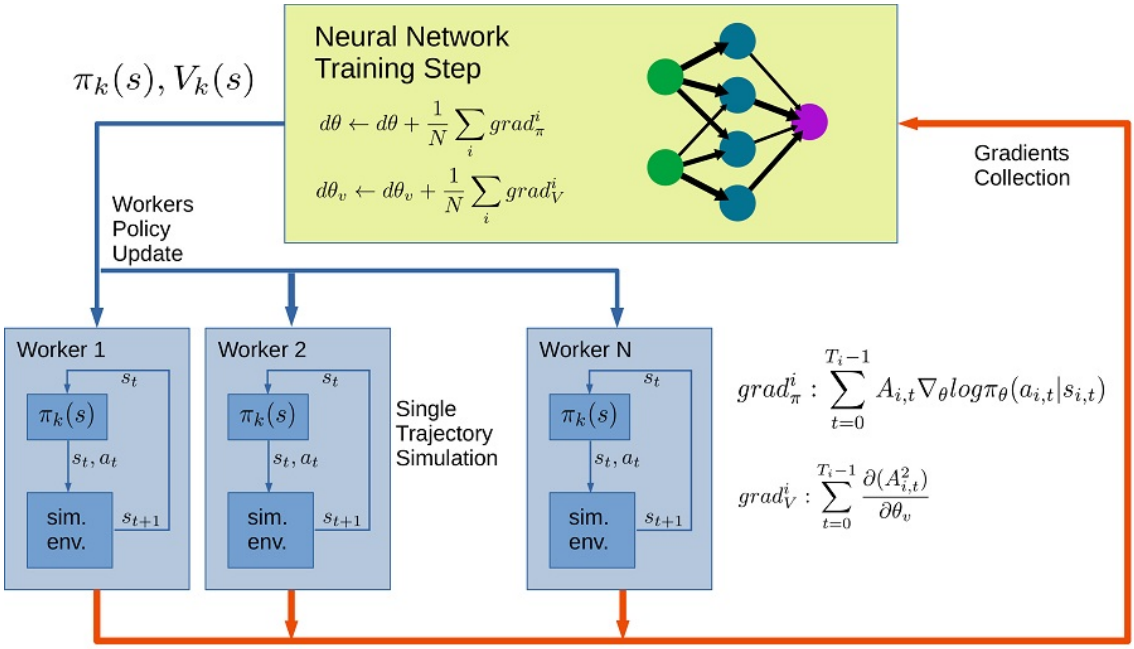


Figure 3.23: A3C algorithm scheme. Worker nodes simulate with action a chosen according to policy $\pi_k(s)$. Gradients are accumulated until a transitions threshold is reached, then NN weights are updated via back-propagation, and policy π_{k+1} is obtained [366].

1. a 3-layer Convolutional Neural Network (CNN) that is responsible for "reading" the data coming from the 2D LiDAR range sensor as well as the robot's linear and angular speed (s_{env});
2. a Fully Connected Neural Network (FCNN) having three hidden layers, which will be referred to as the action branch, which establishes a navigation policy by utilizing the CNN outputs together with the robot state data (s_{rbt}).
3. A second FCNN, from now on called as the *map branch*, which uses the outputs coming from the CNN as inputs and outputs the estimated horizontal separation of the nearest object in each direction. For this reason, the whole 360-degree region surrounding the robot, correctly split again into n_q sectors, is taken into account in addition to the visible portion of the region. It must be pointed out that only during the training process this specific output is utilized, with the mere aim to quicken map reconstruction convergence.

The CNN is an obvious option for the inputs of the measurements obtained from the lidar range sensor, considering its continuous spatial distribution since the sensory input given to the controller of the WMR is three-dimensional with length p, width m, and $n_{ch} = 3$ channels, in which the components are populated with the data in Eq. (3.56). Since the third dimension represents the time variable and is utilized to retain previous observations, the CNN is three-dimensional. Additionally, in the complete

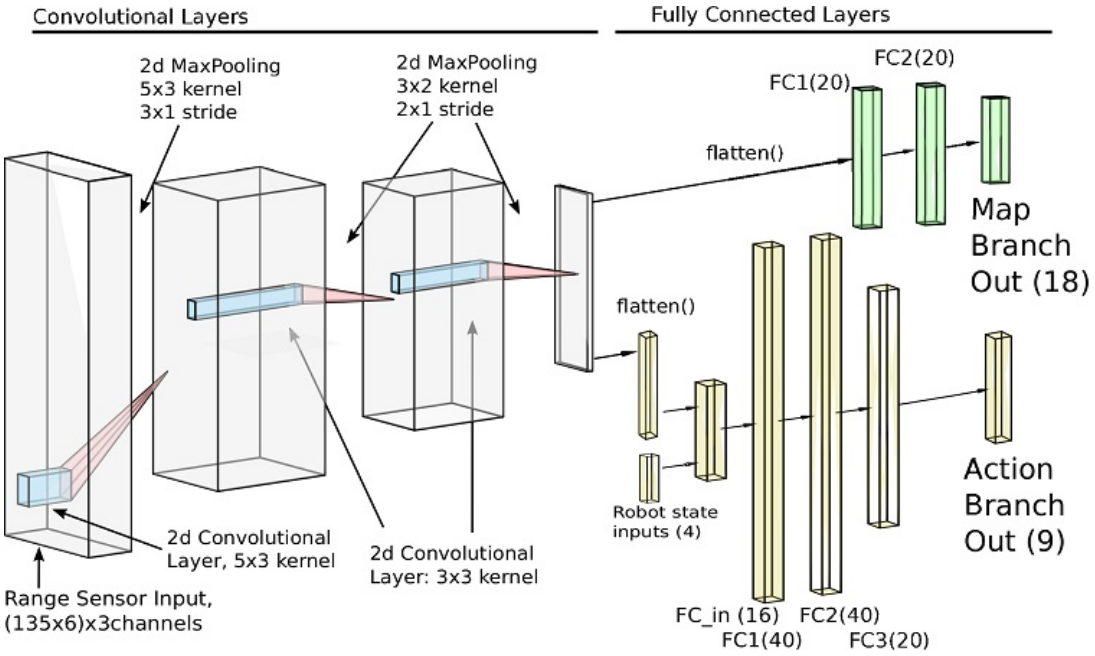


Figure 3.24: Schematic representation of the NN used in the DQL case $Q(s)$. For policy learning $\pi(s)$ a SOFTMAX layer is added at the end, while in the case of the state-value function $V(s)$ the last layer has a unique output [366].

controller's input, the v and ω components are considered to be constant all along the length of the tensor. The normalized input tensor created by s_{env} is convolutioned in two dimensions by the convolutional architecture, which has internal layers with 16 and 10 channels, respectively.

The CNN layer designed in the NN, that has been chosen for this investigation, is composed by:

- a bi-dimensional convolution with a kernel of 5×3 for the initial layer and 3×3 for the subsequent layers;
- an activation function ReLU;
- a bi-dimensional layer MaxPooling having stride of 3×1 for the initial layer and 2×1 for the subsequent layers;
- a layer of normalization along the dimension of the features.

The flattened CNN outcome is transferred to two FCNN's layers with 20 neurons in the map branch, which subsequently outputs the sector-based obstacle distance. As already stated this "map branch" has been utilized only during training. The sector-obstacle distance is therefore expressed as:

$$\hat{d}_i^m, \quad i \in [1, \ldots, n_q] \tag{3.77}$$

Together with the internal robot state s_{rbt}, the CNN outcomes are flattened and subsequently stacked with the former for what concerns the action branch. As a consequence, a FCNN is employed (once more with ReLU

activation functions as well as batch normalization after every intermediate layer), as the new tensor that is produced lacks inherent spatial properties that may be recognized. Finally, the ultimate total of outputs is equal to the total number of actions. Moreover, it must be pointed out that a "softmax" layer is added after the final one whenever the NN is being utilized for policy learning, i.e. when the training has been conducted with the A3C algorithms, whereas for value function $V(s)$, the final layer is made up of a single neuron.

To sum up, the NN may be thought of as two distinct functions that share certain weights and produce the objects $\overline{p}$ and $\hat{d}^m$ from an input/output perspective.

$$\overline{p}, \hat{d}^m = \pi(s_{env}, s_{rbt}) \tag{3.78}$$

with

$$\overline{p} = [p(a_1|s_{env}, s_{rbt}), \dots p(a_{n_a}|s_{env}, s_{rbt})] \tag{3.79}$$

The last thing that needs to be reported, which concerns the topology of NNs, is the addition of an additional loss function, which has been called the "map loss", which was introduced in this study. More precisely, in order to improve the process of learning, the loss functions for RL, i.e. Eq. (3.66) and Eq. (3.73), might be supplemented with additional details. In fact, the dual nature of the information that must be learned in the problem of mobile robots navigating moving crowds, where this dualism can be related to "map" and "strategy", allows for the inclusion of additional information in addition to the rewards. After these considerations, it has been introduced an additional loss function, the "map loss", which needs to be minimized as well, and defined as follows,

$$L_{map}(\theta_i) = \sum_{i=1}^{n_q} (\hat{d}^m_{\theta_i} - d^m_i)^2 \tag{3.80}$$

where by utilizing the map chunk model, which has been illustrated in the problem statement paragraph, the distances d^m_i are obtained. The back-propagation effect with L_{map} makes sure that the CNN learns to use past and current data as quickly as possible to infer the positions of all stationary and moving objects, including those within the shadow cone behind the robot, as precisely as possible.

3.3.4 Training results

In order to facilitate the training process of the NNs, it has been elected, still with the aim to generalize the problem under investigation, to allow for variable densities of pedestrians in the simulated environment. It is apparent that by increasing the density of the pedestrians the harder it will be for the controller to learn the best strategy. It has been elected then to describe the density of the pedestrians in the environment in a discrete way, i.e. levels of pedestrians densities. Subsequently, it has been mapped a problem's difficulty level to the

density levels in such a way that when the difficulty d increases, the pedestrians' density increases as well. From now on, it will be referred to only the difficulty level d, which represents the difficulty of the problem.

This levels distinction has been done because it is apparent that it's not really recommended to perform the training of the NNs directly with the maximum density allowed, but rather it is better to perform the training per difficulty steps, and gradually increase the difficulty of the problem. In Figure 3.25 are illustrated in a graphical form the problem's difficulty levels, i.e. it is possible to see how the difficulty level affects the pedestrians' density in the simulated environment.

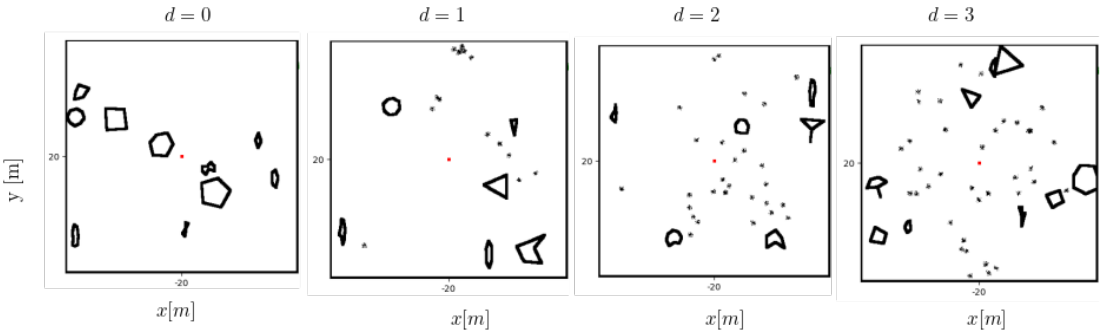


Figure 3.25: Graphical representation of the pedestrian density as a function of the simulation difficulty level d.

The DQL and A3C algorithms' training parameters, that have been chosen to use, enable for comparison of the performances of the RL in terms of reward result and convergence speed. These parameters are listed in Table 3.3.

Table 3.3: Main Simulation hyper-parameters used for the simulated environment and of the training of the NN [366].

Parameter	**DQL**	**A3C**
pedestrians density	0.04 ped/m^2	
scan noise: lost scans	0.5%	
scan noise: corrupt scans	0.2%	
simulating agents	25	
steps per iteration	25000	
minibatch size	256	-
n. epochs per iteration	600	-
optimizer	ADAM	
initial learning rate	$1e^{-5}$	$2e^{-3}$
γ discount factor	0.9	
β (AC entropy coefficient)	-	0.05

However, it is not feasible to directly compare the training simulations due to the setup of the training which has been implemented. In reality, beginning from a random initial condition, then following policy π_t, the A3C training algorithm

creates a certain number of entire trajectories at iteration t. Contrarily, an ϵ-greedy solution which has been outlined in Eq. (3.68), and used for the DQL case, prevents a direct comparison of all the trajectories obtained during the training process. Therefore, for the case of leveraging DQL, it has been elected to simulate one iteration every five by utilizing a pure exploitation strategy as in Eq. (3.64). This has been done in order to conduct a statistical analysis of the results of the simulated trajectories along with the total cumulative rewards.

Both alternatives involve calculating about the same quantity of transitions $s_t, a_t \rightarrow s_{t+1}$ for each training iteration i. A precise number of transitions in each iteration cannot be guaranteed since the trajectories' duration is typically fluctuating. This consideration is apparent if thinking about two cases, in which in the first the robot manages to arrive immediately at the target location, while in the other case, the robot is performing so many evasive maneuvers to avoid pedestrians that it's not reaching the target location and that the simulation ends for a timeout. The training statistics displayed in this section are derived for each iteration on the horizontal axis by averaging the outcomes for all simulation steps/trajectories for both DQL and A3C examples.

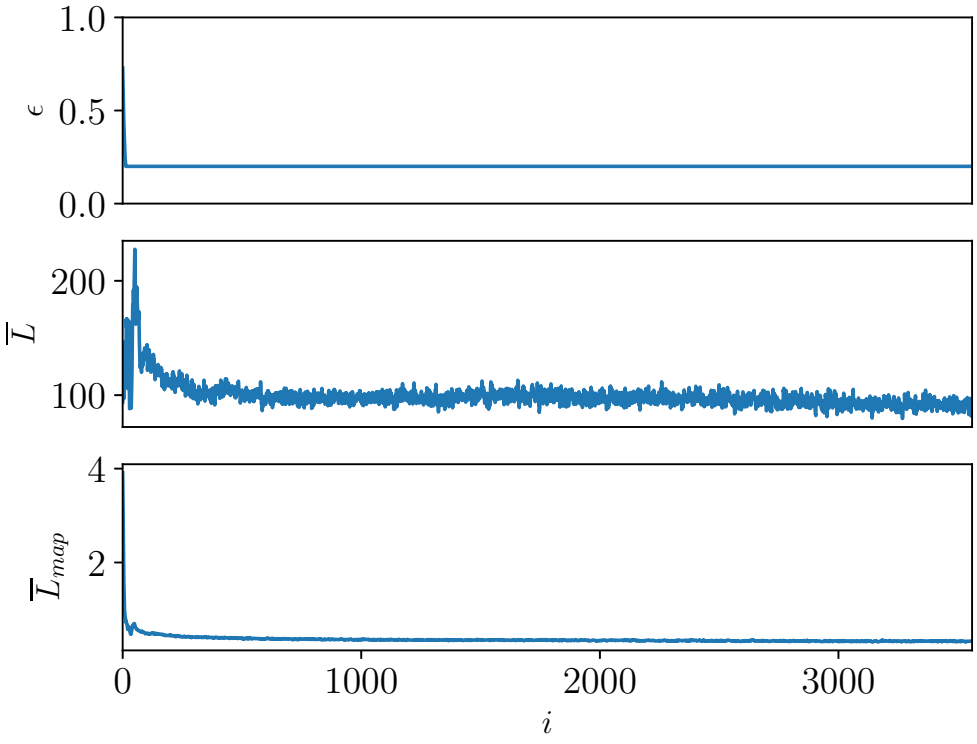


Figure 3.26: DQL training variables evolution: in the top row for ϵ; in the middle row for the average of the q-value loss $\overline{L}$; in the bottom row for the average of the "map loss" $\overline{L}_{map}$ [366].

Figure 3.26 depicts the evolution of the following quantities over the course of the DQL training process: in the top row the parameter ϵ_t of the epsilon greedy method that controls the level of exploration and exploitation; in the second row

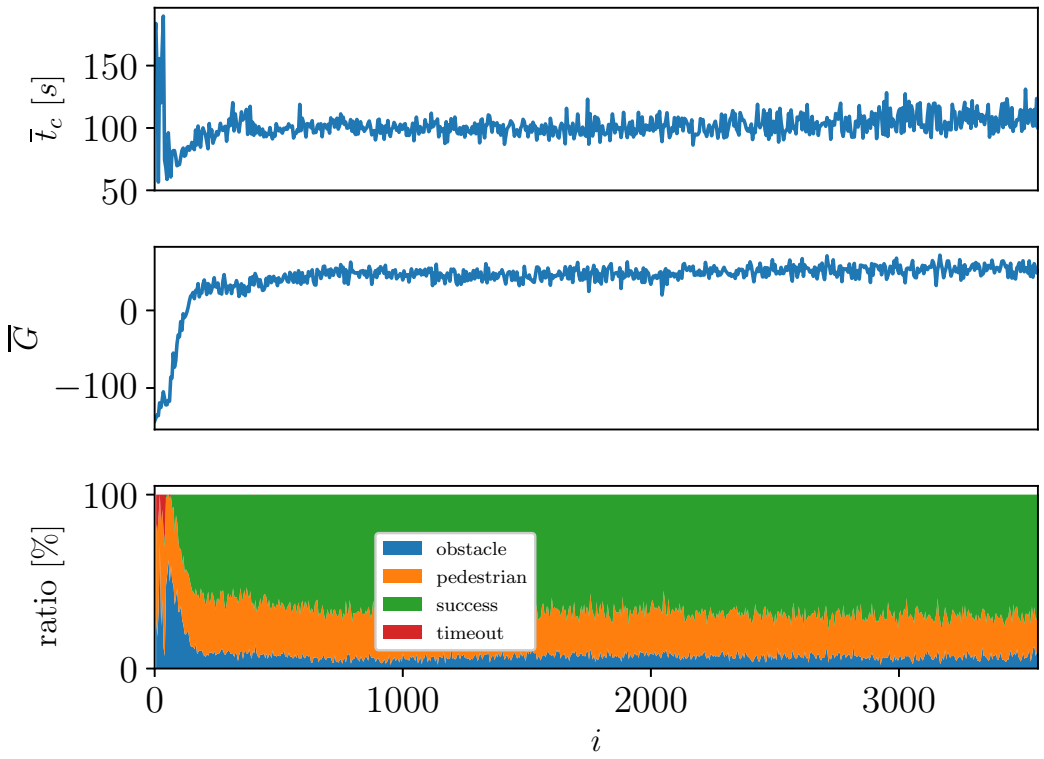


Figure 3.27: Summary of the DQL performance: in the top row the evolution of the average of the duration of the single simulation run $\bar{t}_c$; in the middle row the evolution of the average of the cumulative reward $\overline{G}$; in the bottom row the evolution of the statistic of the simulation termination reason ratio.

the average q-value loss $\overline{L}$, referred to as the mean of Eq. (3.66), on one transition for the current iteration; and in the bottom row the average loss of the "map loss", i.e. the map estimation task, $\overline{L}_{map}$, and defined as the mean of Eq. (3.80). From the same figure, it is apparent that the losses $\overline{L}$ and $\overline{L}_{map}$ converge to a limit value that is unavoidably positive, considering that the 180-degree aperture of the considered range sensor makes it impossible to really "see" half of the map evolution, but can only be estimated.

Still, on the topic of DQL training, Figure 3.27 reports the algorithm's performance metrics. It should be highlighted that only iterations that have been obtained with pure exploitation—for example, one per five—have been taken into account; hence, iterations 1, 6, 11, and so forth. More specifically, the top row displays the evolution of the average simulation length, expressed in seconds, or $\bar{t}_c$; the middle row illustrates the evolution of the cumulative reward, or $\overline{G}$; and the bottom row displays a statistic for the ratio of the reasons which have led the simulation to terminate. Moreover, it can also be observed that the NN learns fairly quickly and that $\overline{G}$, i.e. the average cumulative reward, converges to positive values, however, the NN stops learning around iteration number $i \approx 1000$ since the success ratio appears to be constant and does not show any significant changes. The same trend can be observed from the average cumulative reward

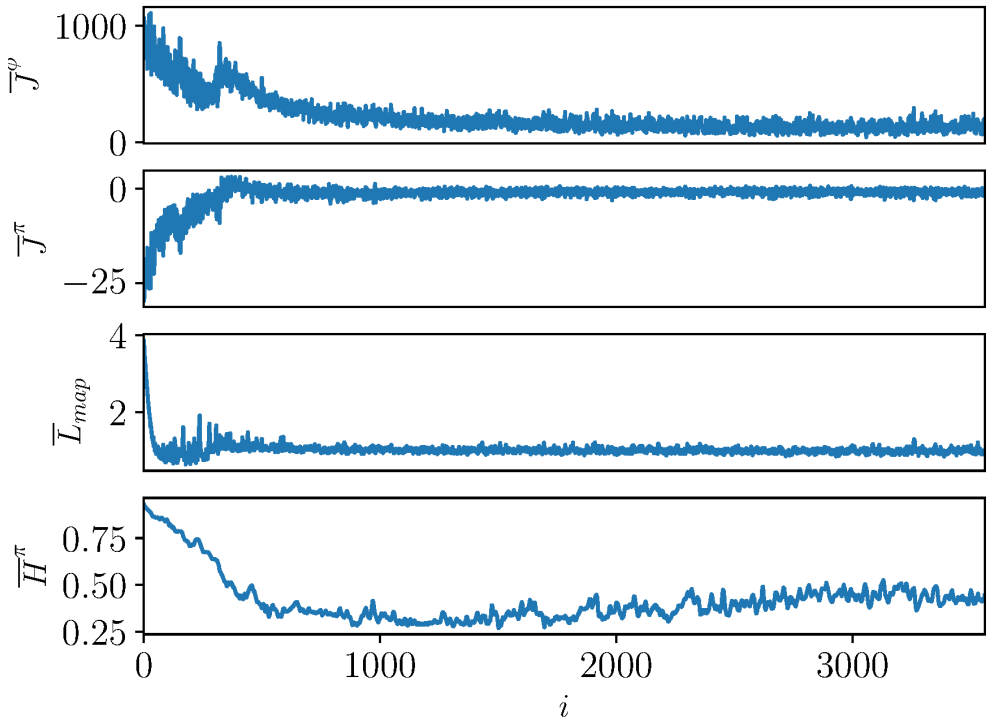


Figure 3.28: A3C training variables evolution: in the top row for the average of the advantage loss $\overline{J}^{\phi}$; in the second row for the average of the policy loss $\overline{J}^{\pi}$; in the third row for the average of the "map loss" $\overline{L}_{map}$; in the bottom row for the average of the normalized entropy $\overline{H}^{\pi}$ [366].

evolution graph.

In contrast, Figure 3.28 and Figure 3.29 display the training results which have been achieved for the NN's training by using the A3C algorithm. Figure 3.28 specifically depicts the evolution of all the key metrics: the average advantage loss $\overline{J}^{\phi}$, referred to as the mean of Eq. (3.74), is shown in the top row, followed by the average policy loss $\overline{J}^{\pi}$, referred to as the average of Eq. (3.71); in the third row is shown the average map loss for the task of estimating the map $\overline{L}_{map}$, and finally the normalized average of the entropy $\overline{H}^{\pi}$, referred to as the average of Eq. (3.75), is shown in the bottom row. As shown, the $\overline{J}^{\phi}$, $\overline{J}^{\pi}$ and $\overline{L}_{map}$ and $\overline{H}^{\pi}$ all converge towards a specific value. The "map loss" is the first of the losses to converge, however, it exhibits convergence to a level that is greater with respect to the DQL example. This is because big batches of training for supervised classification of image-like data perform better.

The performance of the A3C algorithm is presented in Figure 3.29 while still referring to the training procedure that has been carried out by using this method. More precisely, the evolution of the average length of the simulation, expressed in seconds, $\bar{t}_c$ is depicted in the top row; the middle row displays the evolution of $\overline{G}$ together with the filtered curve obtained by applying a moving average filter having window size $w = 100$; finally, a statistic for the ratio of the termination

reason is displayed in the bottom row. Moreover, in this figure, it can be seen that the NN, as opposed to the DQL's trained one, learns a strategy more slowly. However, once it begins to learn, the NN displays a region of constant learning until it reaches convergence; following this, it appears that there is no further significant performance improvement. In spite of this, it is clear that this algorithm has a greater success rate than DQL.

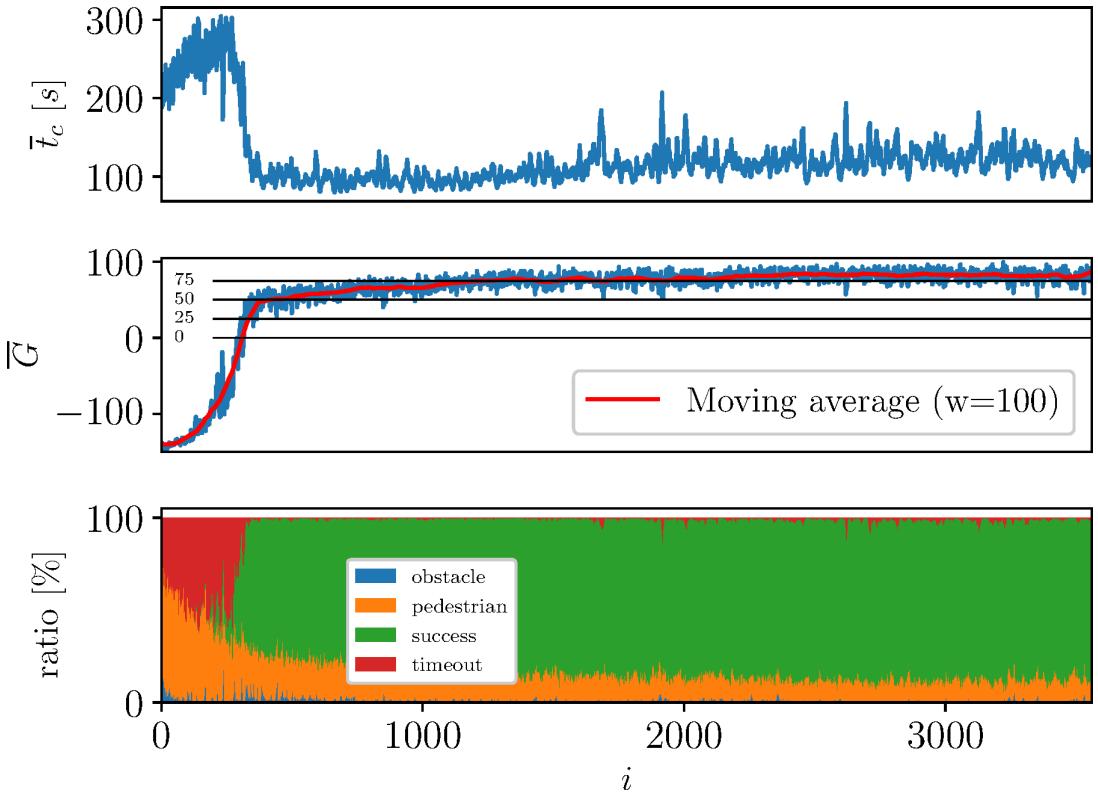


Figure 3.29: Summary of the A3C performance: in the top row the evolution of the average of the duration of the single simulation run $\bar{t}_c$; in the middle row the evolution of the average of the cumulative reward $\overline{G}$; in the bottom row the evolution of the statistic of the simulation termination reason ratio [366].

Finally, a sample A3C algorithm training has been carried out with and without the addition of the extra loss component L_{map}, which has been specified in Eq. (3.80), in order to assess the consequences of its introduction. Figure 3.30 compares the A3C's rate of convergence for the two different scenarios in terms of the average cumulative reward $\overline{G}$ and shows the outcomes. Specifically, when the auxiliary task is used, $\overline{G}$ is represented by an orange curve, and its filtered values are represented by a red line leveraging a moving average filter having a window size of 100. When the auxiliary task is not used, $\overline{G}$ is represented by blue and purple curves.

3.3.5 Testing results

It has been decided to carry out a validation testing campaign in order to evaluate and assess the controllers' performances. The controllers refer to the two NNs

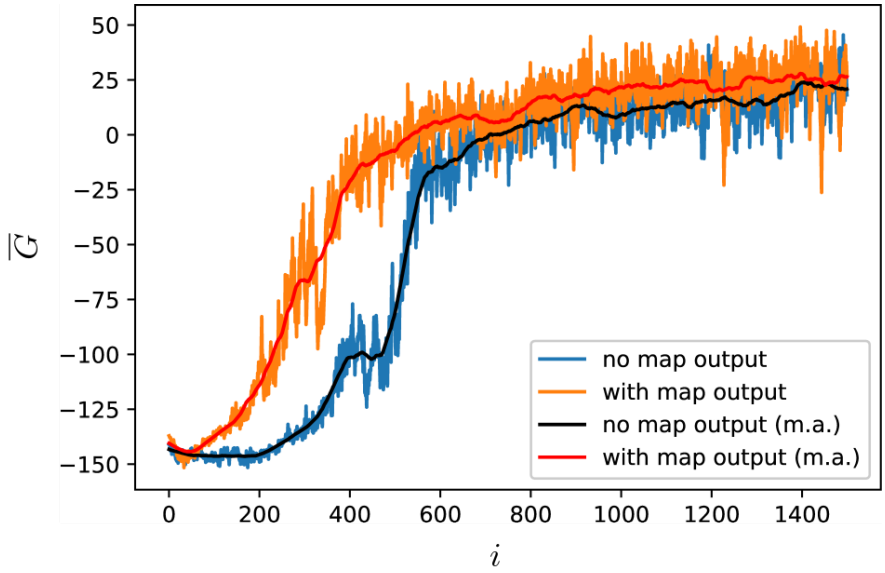


Figure 3.30: A3C Average cumulative reward with and without "map output". First 1500 iterations [366].

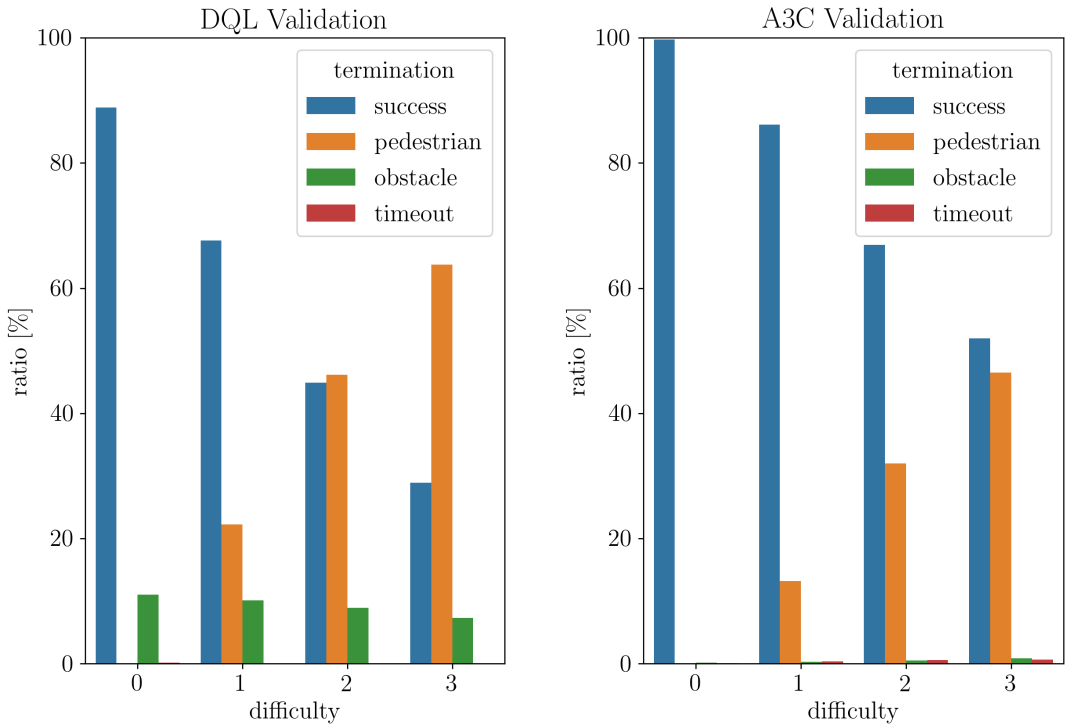
trained using the DQL and the A3C algorithms. More in detail, the trained NN has been loaded into the robot simulation model, where it has been utilized to generate the appropriate actions for the WMR and lead it safely from the starting point to the objective. The version of the NN represented by the most recent iteration, $i = 3560$, has been specifically chosen to be selected for both DQL and A3C cases, with the aim to conduct a proper and cohesive comparison. Furthermore, the testing procedure for both NNs has been assessed for four distinct levels of simulation difficulty, or equivalently the density of individuals inside the simulated environment. Table 3.4 shows the relationship between the level of simulation difficulty and the number of pedestrians present. Additionally, every validation process that has been carried out refers to a mean of approximately $n \approx 10000$ whole simulation runs that have been carried out in simulations.

Figure 3.31 illustrates the outcomes of the testing procedures in the form of bar charts. More in detail, Figure 3.31a shows the success ratio exhibited by the controller trained with DQL as a consequence of the problem's difficulty, or more specifically, the amount of foot traffic in the surrounding area. The single failure percentages, which include the collision with individuals percentage, the collision with static obstacle percentage, and lastly the timeout percentage, are also displayed in the same figure. Additionally, Figure 3.31b depicts the same thing, except in this instance a NN that has been trained with the A3C technique has been employed. While the precise values that have been acquired and graphically displayed in Figure 3.31 are presented in Table 3.5. Finally, both figures present and demonstrate a both qualitative and quantitative comparison of the performances exhibited by both controllers.

As expected, the success ratio for both validated NNs tends to diminish as pedestrian density rises. Due to the problem's increased complexity, the robot is more frequently put in circumstances where it cannot prevent accidents. The

Table 3.4: Mapping between the difficulty levels and the pedestrian density in the environment [366].

Difficulty	Pedestrian density [peds/m^2]
0	0
1	0.02
2	0.08
3	0.1



(a) (b)

Figure 3.31: Barplots indicating the performances of the trained NNs and obtained with the validation process: in (a) the bar plots showing the success, pedestrian collision, obstacle collision, and timeout ratios, respectively for the DQL trained NN; in (b) the bar plots showing the same but for the case of the A3C trained NN [366].

ratio associated with a collision with static obstacles shows a positive trend as pedestrian density rises. This is because the robot attempted evasive movements to escape pedestrians, and as a result, collided with nearby objects. It is clear to see significant distinctions between the performances exhibited by the two trained NNs: at all difficulty levels, the pedestrian collision ratio in the DQL scenario is higher than that in the A3C example. Moreover, regarding the NN trained with the DQL algorithm, it has been noticed that simulations cease for timeout reasons only in the situation of difficulty 0, although with a negligible value equal

to 0.17%. On the contrary, the simulations finish solely as a result of accidents with either static obstacles or individuals in the other three difficulty cases since the timeout termination requirement is never met. The A3C-trained NN, on the other hand, exhibits a certain timeout failure ratio for every level of simulation difficulty. It must be noted, nevertheless, that this percentage associated with this failure does not exceed 1% of cases. Additionally, when comparing the levels of simulation difficulty for the case of obstacle collision ratios, the termination failure percentage exhibits a positive trend. This is probably due to the fact that the robot attempts numerous evasive maneuvers that encourage it to prefer driving until timeout termination criteria is reached over colliding. The A3C-trained NN consistently outperforms the DQL one, demonstrating that policy gradient techniques are more effective at handling this class of challenging issues. As could be seen from the figures, the A3C algorithm can generalize among the many levels of difficulty in a better way. Additionally, the DQL trained NN's less-than-satisfactory performance, particularly at higher difficulty levels, raises the possibility that the training algorithm may have hit a local minimum. This is deduced by the graph in Figure 3.27, in which it can be observed that the NN trained with the DQL algorithm has learned a strategy by training iteration $i = 200$, but the A3C continuously learns and enhances its performance.

Table 3.5: Summary of the results for the validation campaign conducted over the two trained NNs [366].

	Success [%]		Pedestrian [%]		Obstacle [%]		Timeout [%]	
Difficulty	DQL	A3C	DQL	A3C	DQL	A3C	DQL	A3C
0	88.83	99.71	0.00	0.00	11.00	0.17	0.17	0.12
1	67.62	86.13	22.28	13.20	10.10	0.31	0.00	0.36
2	44.92	66.91	46.145	31.96	8.93	0.55	0.00	0.58
3	28.92	52.01	63.75	46.50	7.33	0.85	0.00	0.63

3.3.6 Experimental validation

The outcomes of the carried out experimental validation campaign are provided in this section. It must be pointed out, that it has been chosen to select and employ the A3C-trained NN for all the experimental validation of the controller as well as the approach, since it had generally higher performance, as has been discussed in the previous section. Finally, it has been decided to specifically choose to examine two distinct use case scenarios. These are differentiated in:

- *Case 1. No pedestrians*: In this use case scenario the only obstacles inside the environment are the static ones. The requirement is that the robot must be capable of safely navigating from a starting location $\mathbf{P_i}$ up to a destination point $\mathbf{P_f}$ under the commanded actions received from its trained controller. This requirement can be accomplished either by performing a point-to-point motion, which means that only the terminal point location

is provided, or by performing a path-following motion, which means that a series of way-points, that the WMR must transit through, is explicitly provided.

- *Case 2. With pedestrians*: Within this use case scenario the environment is made up of both static objects as well as moving individuals. Similarly to the previously described scenario, the only requirement asked to the WMR is that it must be capable to arrive at its destination safely while avoiding approaching individuals.

A commercially available Neobotix MP-500, which is a mobile robot featuring differential drive kinematics, is ROS-enabled and is depicted in Figure 3.32, has been chosen to be employed to conduct the experimental validation campaign. The experimental WMR possesses a SICK S300 bi-dimensional LiDAR range sensor, positioned on the front side of the robot. This sensor serves to provide the WMR of the perception system and feeds the NN with its sensor readings, which are required for decision-making, as discussed in the previous sections. Moreover, a suitable ROS package[1]has also been created and therefore installed on the robot's onboard computer in order to allow for control of the robot. Prior to the ROS package being installed on the experimental WMR, it together with the trained NN had been evaluated inside appositely designed scenarios in the dynamic simulation environment of Gazebo [371] to check for any potential flaws or issues. The primary responsibility of the developed ROS package is to collect WMR's sensor data, perform pre-processing on them before feeding them to the input layers of the previously loaded NN, and afterward compute and execute robot speed corrections depending on the actions selected in accordance with the NN policy. The software also offers the following extra functionalities: scanning and mapping of the surroundings, performing SLAM operations (simultaneous localization and mapping), estimation of the robot's pose in the environment, and correction of the robot's odometry-based position, based on environment observations. Furthermore, it enables the final user to designate a specific single point $\mathbf{P_f}$ or even a series of way-points that the WMR must travel through. The user is facilitated because this happens by using the visualization software *RViz* GUI. Otherwise, it enables the final user to load waypoints from a previously defined file. It must be pointed out that the developed ROS package has been created to be universal and can thus be used with both experimental WMR and the simulated model within Gazebo dynamic simulator.

Gazebo Pre-Testing

As mentioned above, the ROS package has been developed in order to be used either in the experimental setting or the simulated environment Gazebo, without any modification of the source code. This aspect makes it very useful in order to perform fine-tuning of the controller and check the correct functioning of the mobile robot in the simulated environment before deploying it on the real hardware.

[1]The data and the source code are publicly available at the following GitHub repository `https://github.com/matteocaruso1993/crowd_nav_experimental`.

Figure 3.32: The elected differential drive mobile robot used in the experimental campaign: a Neobotix MP-500 [366].

With these considerations it has been decided to perform some ROS-Gazebo co-simulations, to check the functioning of the various systems composing the whole software of the robot, and evaluate the global behavior of the WMR. The whole software shown as a ROS nodes connectivity graph is shown in Figure 3.33. It must be pointed out that the ROS nodes connectivity graph doesn't change from the one representing the software of the experimental robot.

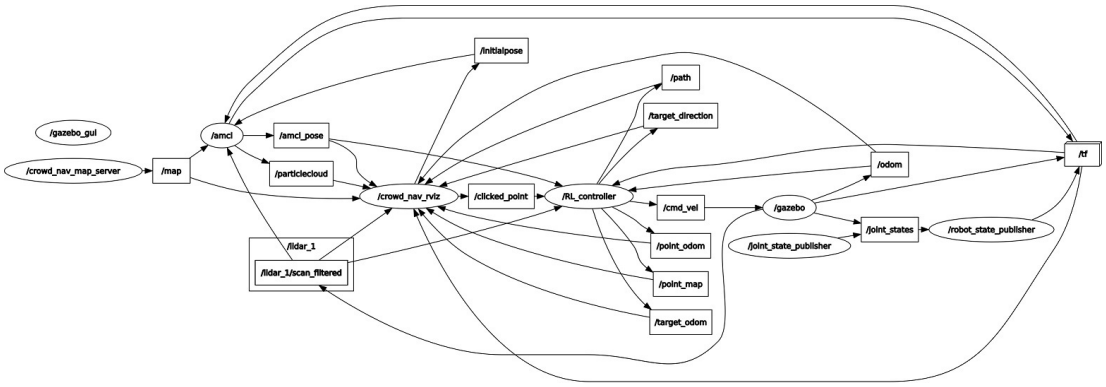


Figure 3.33: Rosgraph of the ROS-Gazebo co-simulation application for a mobile robot navigating moving crowds.

Within Gazebo it has been modeled an environment which is composed of a large squared room having size $20 \times 20\text{m}^2$, in which are posed some obstacles having simple shape, and walls are placed on each side of the room. Within this environment, it has been tested only: the capability of the mobile robot to reach a commanded target location, and the capability to detect and avoid static obstacles. Therefore, no dynamic obstacles are implemented in this stage. The ROS package, for the case of Gazebo simulations, is responsible for starting the Gazebo simulator with the selected environment, spawning the Neobotix MP-500 mobile robot in the environment, performing environment mapping, loading and

assigning to the robot the RL trained controller, starting the simulated sensors, starting the positioning sub-system and starting the RViz visualization tool.

Subsequently, the mobile robot has been spawned on one side of the room, while the target location has been chosen to be placed on the other side of the room. The time-series evolution of the performed simulation can be seen in Figure 3.34, while the top view of the environment and the trajectory performed by the mobile robot can be seen from Figure 3.35.

Figure 3.34: Time-serie evolution of the ROS-Gazebo simulation

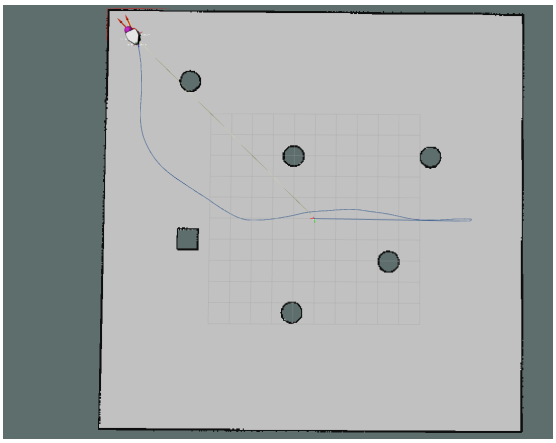


Figure 3.35: Map of the Gazebo simulation environment and trajectory followed by the mobile robot seen from top view in RViz.

From those figures, it can be seen that the mobile robot successfully reached the commanded target location. Moreover, it is clear that the WMR has been capable of detecting obstacles and correcting its heading to avoid them. Finally, after testing the controller in the Gazebo simulated environment and fine-tuning the controller, it has been deployed the RL trained controller on the real experimental mobile robot and performed the experimental validation tests on a real setting.

Case 1. No pedestrians

It has been decided to carry out the experimental validation testing for both the motions point-to-point as well as via way-points under this case. The University of Trieste's C6 lab, an indoor setting, is where the experiments had been specifically carried out.

By utilizing the mapping capabilities offered by the developed ROS package at the experimental testing setup, the so-called "occupancy grid", i.e. a bi-dimensional map, of a portion of the laboratory setting had been created and can be seen from Figure 3.36. Specifically, numerous static obstacles may be seen on the map that had been reconstructed. Unfortunately, in the environment, there are a few obstacles that the range sensor was unable to detect. The test scenario only for point-to-point motion can be seen in Figure 3.36. In this scenario, the WMR is required to navigate up to the given target point (represented by the red "TARGET" label) picked to be situated in the bottom-right corner of the laboratory, starting from the top-left corner of the laboratory setting (marked by the green "START" label). The position of the target point had been set such that the WMR must navigate around numerous obstacles including desks, cabinets, and machinery in addition to performing a 90-degree turn and navigating through a small passageway just before the target site. Additionally, the WMR is required to safely avoid any static obstacles that are in its path as it moves inside the lab, i.e. has to perform obstacle negotiation. In the meanwhile, the WMR's performed trajectory is depicted by the blue line. The WMR is claimed to have completed the specified task of performing point-to-point motion tasks successfully, as well as avoiding the static obstacles that it has found on the way. Additionally, it is also evident that the WMR moves in such a way that it maintains almost the same safety distance from surrounding obstacles while it is in motion.

For the case of the motion through way-points, the target point has been put on the opposite side of the lab from where it had been during the previously carried out validation test since it has been decided to permit the WMR to travel a definitely longer course. This is evident in Figure 3.37, where the "START" green label and the "TARGET" red label, respectively, designate the initial and target coordinates. Furthermore, the allocated waypoints are identified by the black "WAYPOINTS" label, whereas the WMR's path is still represented by the blue curve. Considering the same picture, it should be noticed that the WMR spends the majority of its motion traveling through a relatively tight hallway.

Case 2. With pedestrians

The laboratory has once again been chosen as the indoor testing site for this last experimental validation test. In contrast, the robot in this scenario must carefully avoid moving individuals which cross its pathway as well as traveling nearby in addition to performing a point-to-point motion and dodging static objects, as has been required for the previous cases. Moreover, the initial point and target point provided to the WMR, which are indicated in Figure 3.38, are the same ones that have been selected for the point-to-point use case scenario and mentioned in the preceding paragraph. These points are again shown in green and red, respectively.

Figure 3.36: Trajectory of the mobile robot for the point-to-point motion in the absence of dynamic obstacles [366].

Throughout this point-to-point motion, the mobile robot's performed course is once more shown with a blue curve.

The first thing that can be observed from Figure 3.38 is how the curve that represents the WMR's performed trajectory has tangles in certain spots. This phenomenon is due to the excessive proximity of a moving individual to the areas where the WMR had been operating; as a result, the latter had been forced to execute an evasive maneuver to prevent colliding with the former. It should be highlighted that the WMR had managed to avoid simultaneously static obstacles as well as moving individuals while simultaneously traveling up towards the specified destination fully safely and without any accidents, despite the complexity that had been added just by the introduction of moving individuals in the testing setting.

The outcomes of this conducted experimental validation testing also imply that a straightforward bi-dimensional LiDAR scanner is capable of detecting individuals' legs and is appropriate for crowd navigation applications, thus also confirming that the ray-casting approach introduced in the simulation environment is a reasonable choice. This has resulted in the reduction of the so-called "reality gap" between the simulated scenarios and the real ones.

Last but not least, Figure 3.39 displays a few video frames that had been

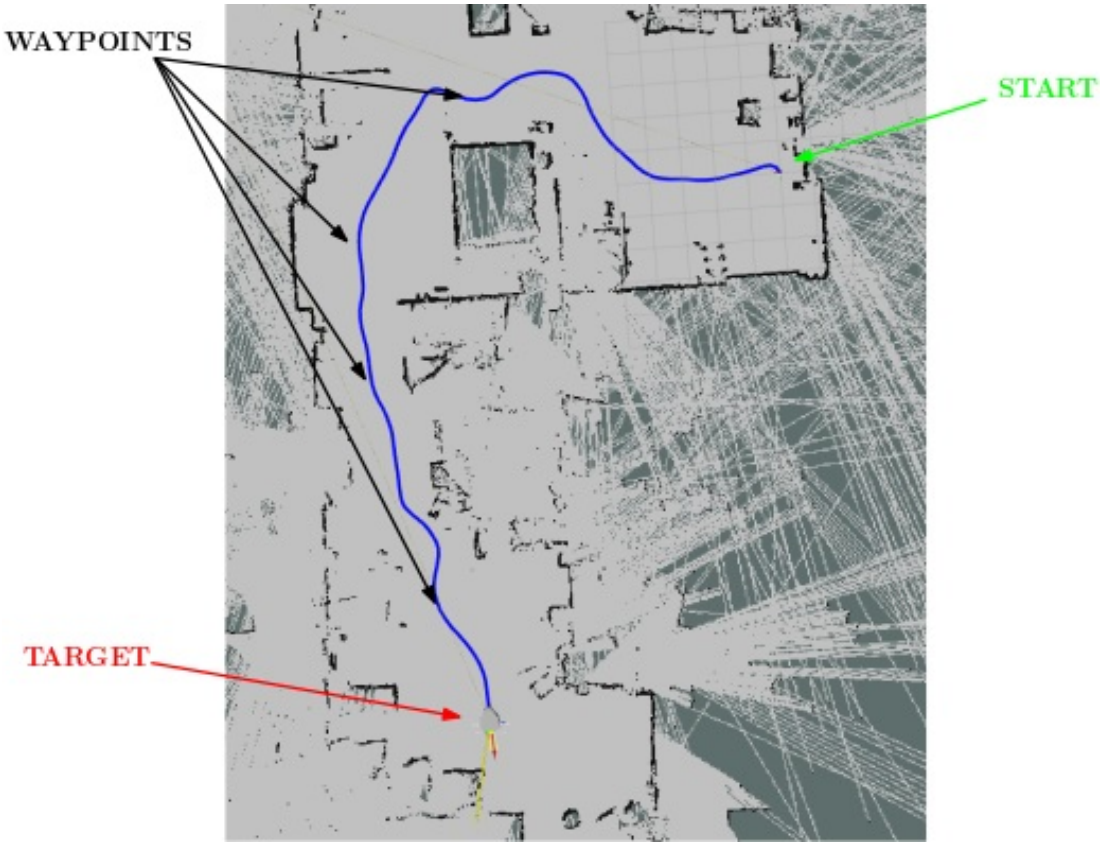


Figure 3.37: Trajectory of the mobile robot for the path following through given way-points in the absence of dynamic obstacles [366].

taken from the WMR's camera. The frames refer to the case of point-to-point motion case with moving pedestrians. The latter may be seen entering the route of the WMR and being in close proximity to it in a few of the frames.

Figure 3.38: Trajectory of the robot for the point-to-point motion in presence of moving pedestrians as dynamic obstacles [366].

Figure 3.39: Extracted video frames from the robot camera during its point-to-point motion in presence of moving pedestrians [366].

Chapter 4

Conclusions

In this dissertation, several topics and issues involving mobile robots were studied and addressed. These issues involved mobile robots and their applications in two main declinations: mobile robots for applications in the context of space exploration and mobile robots for applications in industrial settings. In addition, in all of these topics particular attention was paid to the identification of the interactions that the robotic system under consideration undergoes. These were therefore considered, carefully modeled, and finally addressed. Specifically, the study and design of a robotic lander were covered, and then the problem of soft-landing associated with it was addressed. The study and design of a prototype planetary rover Archimede completely based on the ROS framework was then explored. Modeling of the rover was performed and the dynamics of the system subject to impacts and driving tests were studied. The problem of manipulating ground-laid objects through the towing technique was also seen and then applied to the Archimede rover subjected to constraints on the steering axes. The problem of developing a fleet manager for a system of tethered robots, by exploiting a kinematic analogy with redundant robots was analyzed, and finally the methodology was applied and studied for different industrial application contexts. Finally, the problem of crowd navigation was introduced and studied, and the development of a controller, trained with RL techniques, was carried out for a mobile robot that safely navigates the moving crowd. For each of these applications a small paragraph is devoted to them in this chapter.

Summarizing, the main achievements and results obtained in this dissertation can be summarised as follows,

- For the space exploration robotics field: the control of vehicles for space exploration in exotic applications, e.g. manipulation of objects via towing, and the study of the dynamics of complex robotic systems such as articulated planetary rovers and landers.

- For the mobile robotics for industrial settings applications: the development of a framework for the synthesis of a fleet manager that controls a fleet of tethered robots, and the development of a framework and pipeline for the development of smart controllers for mobile robots navigating moving crowds.

In Sec. 2.2 the use of a passive shock absorber in robotic landers for space applications has been studied. Specifically, a novel three-legged robotic lander has been studied and designed, based on the usage of VRDs, cables, and preloaded springs. The behavior of the lander has been numerically investigated in order to demonstrate the efficacy and soundness of the developed analytical model for the dynamics of the robotic lander. First, the analytical model was implemented, then the MBS simulator ADAMS was used on a real-world 3D CAD model of the proposed lander. It is possible to conclude that the analytical model accurately depicts the behavior of the lander for the situation of soft-landing on a horizontal plane given the similarities between the findings obtained by using the MATLAB-based analytical model and the findings acquired in the ADAMS simulator. However, it has been noted that the analytical model overestimates the contact force as well as the acceleration that the lander experiences at impact with regard to ADAMS. There could be a number of causes for this mismatch. The two key ones are that – in contrast to the analytical model – first, the ADAMS model incorporates the three ratchet devices; and second, that the contact point with the soil changes due to the curvature of the landing pad during leg rotation. Additionally, the considered contact models differ, with the analytical model using a point-to-point contact model and ADAMS using a solid-to-solid contact model. Additionally, for contacts, the analytical model uses a linear damper, whereas ADAMS uses a non-linear one, hence translating into the fact that the dissipative force generated by the models in the phase of impact differs. Finally, it has been noted that in the simulations, this type of passive landing mechanism ensures a certain level of control over the forces and accelerations exerted on the structure of the robotic lander during soft-landing operations.

In Sec. 2.3, by leveraging sophisticated compliant joints known as S-Structures, Archimede is proposed, a unique four-steerable wheel space exploration planetary rover prototype. The rover, completely based on ROS, has been described in every single component. Specifically, an approach based on the projection of the Instantaneous Center of Rotation is developed in order to provide a stable methodology to control a vehicle subjected to wheel steering constraints. Moreover, it has been presented the development of an odometry subsystem for position estimation of the rover. The modeling of the Archimede rover and all its systems in Gazebo has been presented. The odometry subsystem has been demonstrated both in simulations and experimentally, and its accuracy has been evaluated. Using Kane's technique, a semi-analytical (SA) model of the rover dynamics is developed. The model aims at reducing the complexity of the robotic system by encapsulating the intricate S-Structure mechanism in a straightforward revolute joint characterized by having a preloaded non-linear elastic response. Two further distinct models are offered in this section: an experimental prototype and an ADAMS-based model. Despite the numerous sources of inaccuracy, results from numerical and experimental comparisons of all models have revealed that there is good conformity between the models, particularly between the SA and the experimental. It becomes clear from the analysis of the results and also from the comparison of the different models, reported in Sec. 2.3.11, how using the SA model may be advantageous in terms of precision when compared to reality and

also in terms of computation complexity; in fact, the SA model is significantly less sophisticated with respect to the one that has been developed in ADAMS. In fact, the latter has significantly more moving parts – hence DOFs – as well as internal contacts. For these connections in particular to be representative, thorough measurements of the physical counterpart real contact mechanics are necessary. The SA technique allows to efficiently combine these intricate effects into two parameters, c_q and f, which makes the model more adaptable and simple to use.

In Sec. 2.4 the potential for a mobile robot to manipulate a body, that is lying on the ground, using a tether has been explored. The manipulation strategy that has been considered is the towing approach. It has been suggested to represent the distribution of the contact pressure in a formal and tasteful manner. The kinematics and joint restrictions introduced by the Archimede rover prototype as the towing actor, have been taken into consideration in order to synthesize both the object and rover trajectories. The performed dynamic simulations highlight the importance of the main model hyper-parameters as described in the thorough sensitivity analysis. The investigation of the path-following performance has revealed information about the flexibility of the path-planner, and a small number of trajectories has been chosen. The physical Archimede rover prototype, which has steerable wheels, was utilized in an experimental campaign. An ad-hoc image tracking software package has been used to examine the motion of the towed object as well as the Archimede rover in order to assess the introduced models, i.e. the dynamics, friction, and contact ones. Results demonstrate the models good realism. It had been possible to simulate the behavior of the rover Archimede towing an object of any form and a non-constant distribution of the contact pressure across a hard surface using the methodology developed within this study. It has been then demonstrated how this type of modeling can be used effectively to conduct path-planning for the towed object. In addition, it has been demonstrated how the strategy is influenced by the numerous parameters and specific hyper-parameters of the models and how these influences affect how closely the towed body follows the planned path.

In Sec. 3.2 it was demonstrated how a kinematic analogy between a chain of tethered mobile robots (TMR) and a redundant robotic manipulator can be used to control the mobile robots. Indeed, these can be exposed to to the task-related constraints of a redundant robotic manipulator and, by making use of its redundant degrees of freedom, they can be effectively controlled. The constraints in this application are: avoiding hazards, maintaining a safe distance between the robots, maintaining the separation between adjacent robots below the maximum allowed by the catenary-described cable element, decreasing the space between them, and maintaining this distance uniform amongst the robots. The approach occasionally showed limited erratic motion; for instance, whenever the obstacle avoidance auxiliary task activates to avoid a nearby obstacle, it occasionally results in an abrupt lateral displacement. It is significant to note that the omnidirectional motion capabilities of mobile robots have been taken into account in this study; this allows viable lateral motions along the planned trajectories, which is advantageous. On the other hand, more caution should be paid if the mobile

robots are not omnidirectional. In that instance, this can cause the tracking error relative to the robots' nominal trajectories to grow. The solution of the TMR path planning exhibits numerical disturbances, which, according to the results, can be reduced with minor post-processing on the generated trajectories. It is believed that by allowing a smooth transition between the distinct tasks, such as through a blending function rather than the current abrupt transition, these oscillations might be decreased, as is done in earlier publications [317, 372, 373]. Additionally, the sensitivity analysis revealed that the d_{min} parameter and the obstacle avoidance auxiliary task, in general, have a significant impact on the considered objective functions, which lowers the robots performances as evidenced in the energy and jerk-related functions. The same demonstrates that it is possible to configure the chosen parameters near the failure configuration regions and yet achieve the best behavior of the entire system.

In Sec. 3.3, the development has been investigated of a controller which allows a mobile robot to navigate to target locations while avoiding both static and moving objects, such as pedestrians. The mobile robot controller relies on the employment of a Neural Network (NN) that has been trained using the DQL and A3C Reinforcement Learning (RL) algorithms. Comparisons between trained NNs indicate that the A3C-trained NN performs better and generalizes the issue more successfully than the DQL-trained NN. While both techniques are capable of achieving convergence during training, it has been demonstrated also that the NN trained with DQL learns a sub-optimal approach. The training time for this type of complex problem was drastically reduced by utilizing the developed pipeline to perform the training procedure on the NNs in a highly parallelized environment. In fact, without parallelization, the training time would make the problem very challenging to solve. Furthermore, it has also been demonstrated that NN trained using reinforcement learning approaches can successfully handle difficult problems like crowd navigation. In fact, it has been shown that the robot is able to effectively execute the specified task with a great success ratio for varying environment pedestrian densities in the simulated environment utilizing both trained NNs. The method has also undergone experimental validation, during which it was discovered that the considered mobile robot exhibits excellent obstacle avoidance abilities and is capable of traveling between way-points and conducting point-to-point mobility in the presence or absence of both dynamic and static obstacles. Additionally, it has been demonstrated that a straightforward perception system, such as a LiDAR range sensor, may be utilized to accurately recognize pedestrian legs and subsequently deliver useful information to the input layer of the NN. Ray-casting methods are thus used in the simulated surroundings to accurately reflect the sensor operating principle. The validation of the approach has been done experimentally by using and deploying the A3C-trained NN. It should be noted that the testing environment for the controller and the robot was not very "robot friendly" and presented several difficulties for the mobile robot as well as its perception system. In fact, the environment is not well-organized and contains several impediments that the robot perception system finds challenging to recognize. More specifically, there are obstacles that are too high, too low, or too small to be seen, such as chairs and desk legs. Nev-

ertheless, despite these shortcomings, the mobile robot was nevertheless able to carry out the given duty in a safe and efficient manner.

In conclusion, it has been shown in this dissertation how the interactions of mobile robots with their surroundings, with other robots, and finally with humans are of paramount importance for proper modeling of case studies. As it has been seen, depending on the application contexts, a generic mobile robot is subjected to a combination of all the aforementioned interactions, but not all of them assume the same relevance for the given case study. These interactions must be modeled with a sufficient level of accuracy and detail in order to best describe the problems studied, and they cannot be neglected a priori. This has been observed in all the studies addressed and presented in the previous chapters, specifically applied in different contexts and problems, especially for the case of navigating a mobile robot within a moving crowd. The techniques studied and developed in this dissertation show how some of these aspects can be modeled and thus predicted, providing the tools to address the problems, of particular scientific interest in the field of mobile robotics, to the best of one's ability.

Finally, this dissertation work explores topics of global interest in space exploration and industrial robotics and builds a good foundation on which to build future research. Some possible real world applications, especially focused on space exploration and industrial fields, have been seen throughout this dissertation. However, some other alternative possible real world applications of the developed methodologies can be identified. For example, it is possible to use the object manipulation strategy presented in Sec. 2.4 for planning load handling operations within industrial warehouses. Furthermore, due to the nature and the general definition given to the problem of mobile robots navigating moving crowds provided in Sec. 3.3, it is thought that the developed methodology can also have repercussions in other contexts, other than the mere context of crowd navigation. Indeed, due to the formulation given to the problem, the long distances traveled by the robot, and the independence from the specific simulation environment, extension to other contexts, such as the ones of the e-commerce and home delivery, is feasible. Another relevant application of this methodology is thought to be the use of these smart controllers on autonomous mobile robots operating inside industrial warehouses and in manufacturing contexts, for navigating environments populated by humans and other robots simultaneously, during movement within and between rooms. Finally, although the Archimede rover, presented in Sec. 2.3, is designed and developed for purely space and planetary exploration applications, some of its concepts and systems can be applied to other contexts that are not necessarily space-based. For example, it uses a suspension system designed for fast driving over rough, granular, non-consistent terrain, and which provides some degree of resistance to shocks and impacts. It is thought, for example, that the use of this type of system could also be used on robotic devices designed for use in robotic agriculture, outdoors, and generally unstructured environments. In addition, through the use of this suspension system, combined with machine learning techniques and appropriate smart controllers, e.g. the one developed for crowd navigation purpose, it is thought to extend the operational contexts of mobile robots into previously inaccessible scenarios, e.g.,

the ability of mobile robots to navigate very complex and varied environments such as forests.

Bibliography

[1] J. Buchli, *Mobile Robots - Moving Intelligence*, ser. ARS, Advanced robotic systems international. Pro-Literatur-Verlag, 2006.

[2] F. Wenzhöfer, J. Lemburg, M. Hofbauer, S. Lehmenhecker, and P. Färber, "Tramper - an autonomous crawler for long-term benthic oxygen flux studies in remote deep sea ecosystems," *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6, September 2016.

[3] L. E. Bruzzone and G. Quaglia, "Review article: locomotion systems for ground mobile robots in unstructured environments," *Mechanical Sciences*, vol. 3, pp. 49–62, 2012.

[4] Y. Zhao, X. Chai, F. Gao, and C. Qi, "Obstacle avoidance and motion planning scheme for a hexapod robot octopus-iii," *Robotics and Autonomous Systems*, vol. 103, pp. 199–212, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889017303329

[5] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, ser. Mechanical Engineering Series. Springer US, 2008. [Online]. Available: https://books.google.de/books?id=3yOUzgEACAAJ

[6] M. Raibert, "Bigdog, the rough-terrain quadruped robot," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 17, no. 1 PART 1, 2008.

[7] D. Seidel, M. Hermann, T. Gumpert, F. C. Loeffl, and A. Albu-Schaffer, "Using elastically actuated legged robots in rough terrain: Experiments with dlr quadruped bert," in *IEEE Aerospace Conference Proceedings*, 2020.

[8] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots, second edition*, ser. Intelligent Robotics and Autonomous Agents series. MIT Press, 2011. [Online]. Available: https://books.google.de/books?id=4of6AQAAQBAJ

[9] K. Tadakuma, R. Tadakuma, A. Maruyama, E. Rohmer, K. Nagatani, K. Yoshida, A. Ming, M. Shimojo, M. Higashimori, and M. Kaneko, "Mechanical design of the wheel-leg hybrid mobile robot to realize a large wheel diameter," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, p. 3358 – 3365.

[10] N. Babu, S. Sujatha, S. Narayanan, and V. Balamurugan, "Novel hybrid leg-track locomotion robot and its stability analysis using a unified methodology," *Procedia Computer Science*, vol. 133, pp. 486–493, 2018, international Conference on Robotics and Smart Manufacturing (RoSMa2018).

[11] P. Ben-Tzvi and W. Saab, "A hybrid tracked-wheeled multi-directional mobile robot," *Journal of Mechanisms and Robotics*, vol. 11, no. 4, 2019.

[12] L. Bruzzone, M. Baggetta, S. E. Nodehi, P. Bilancia, and P. Fanghella, "Functional design of a hybrid leg-wheel-track ground mobile robot," *Machines*, vol. 9, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2075-1702/9/1/10

[13] B. Klaassen and K. L. Paap, "Gmd-snake2: a snake-like robot driven by wheels and a method for motion control," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 4, 1999, p. 3014 – 3019.

[14] A. D. Horchler, A. Kandhari, K. A. Daltorio, K. C. Moses, K. B. Andersen, H. Bunnelle, J. Kershaw, W. H. Tavel, R. J. Bachmann, H. J. Chiel, and R. D. Quinn, "Worm-like robotic locomotion with a compliant modular mesh," in *Biomimetic and Biohybrid Systems*, S. P. Wilson, P. F. Verschure, A. Mura, and T. J. Prescott, Eds. Cham: Springer International Publishing, 2015, pp. 26–37.

[15] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, "Model predictive robot-environment interaction control for mobile manipulation tasks," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, 2021, p. 1651 – 1657, all Open Access, Green Open Access.

[16] B. Hichri, J.-C. Fauroux, L. Adouane, I. Doroftei, and Y. Mezouar, "Design of cooperative mobile robots for co-manipulation and transportation tasks," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 412–421, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584518301339

[17] H. Roth, K. Schilling, and O. J. Rösch, "Haptic interfaces for remote control of mobile robots," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 177–182, 2002, 15th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015393575

[18] N. C. Mitsou, S. V. Velanas, and C. S. Tzafestas, "Visuo-haptic interface for teleoperation of mobile robot exploration tasks," in *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, 2006, pp. 157–163.

[19] D. Gêgo, C. Carreto, and L. Figueiredo, "Teleoperation of a mobile robot based on eye-gaze tracking," in *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, 2017, pp. 1–6.

[20] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space," in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.

[21] M. C. Shrestha, T. Onishi, A. Kobayashi, M. Kamezaki, and S. Sugano, "Communicating directional intent in robot navigation using projection indicators," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2018, pp. 746–751.

[22] J. Berg, A. Lottermoser, C. Richter, and G. Reinhart, "Human-robot-interaction for mobile industrial robot teams," in *Procedia CIRP*, vol. 79, 2019, p. 614 – 619, all Open Access, Gold Open Access.

[23] K. Schilling and C. Jungius, "Mobile robots for planetary exploration," *IFAC Proceedings Volumes*, vol. 28, no. 11, pp. 109–119, 1995, 2nd IFAC Conference on Intelligent Autonomous Vehicles 1995, Espoo, Finland, 12-14 June 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017469588

[24] L. Yenilmez and H. Temeltas, "Autonomous navigation for planetary exploration by a mobile robot," in *International Conference on Recent Advances in Space Technologies, 2003. RAST '03. Proceedings of*, 2003, pp. 397–402.

[25] A. M. Gargiulo, I. di Stefano, and A. Genova, "Model-based slippage estimation to enhance planetary rover localization with wheel odometry," *Applied Sciences*, vol. 11, no. 12, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/12/5490

[26] E. T. Baumgartner, H. Aghazarian, A. Trebi-Ollennu, T. L. Huntsberger, and M. S. Garrett, "State estimation and vehicle localization for the FIDO rover," in *Sensor Fusion and Decentralized Control in Robotic Systems III*, G. T. McKee and P. S. Schenker, Eds., vol. 4196, International Society for Optics and Photonics. SPIE, 2000, pp. 329 – 336. [Online]. Available: https://doi.org/10.1117/12.403731

[27] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 903–910 Vol. 1.

[28] J. H. Carrió, J. Schwendner, and F. Kirchner, "Planetary rover localization design: Antecedents and directions," in *2012 IEEE Intelligent Vehicle Symposium*, Madrid, Spain, 03.6.-07.6.2012.

[29] Y. Zhan, Y. Zheng, C. Li, R. Wang, Y. Zhu, and Z. Chen, "High-accuracy absolute positioning for the stationary planetary rover by integrating the star sensor and inclinometer," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1063–1076, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21944

[30] F. Ingrand, S. Lemai-Chenevier, and F. Py, "Decisional autonomy of planetary rovers," *Journal of Field Robotics*, vol. 24, no. 7, p. 559 – 580, 2007, cited by: 37; All Open Access, Green Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-34547893113&doi=10.1002%2frob.20206&partnerID=40&md5=6810f039a37683f3f59d371559b50637

[31] W. Wu, X. Xie, M. Wei, N. Liu, X. Chen, P. Yan, M. Omar, and L. Xu, "Planetary rover path planning based on improved a* algorithm," in *Intelligent Robotics and Applications*, H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, and D. Zhou, Eds. Cham: Springer International Publishing, 2019, pp. 341–353.

[32] A. K. Guruji, H. Agarwal, and D. Parsediya, "Time-efficient a* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144–149, 2016,

3rd International Conference on Innovations in Automation and Mechatronics Engineering 2016, ICIAME 2016 05-06 February, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017316300111

[33] J. Guo, L. Liu, Q. Liu, and Y. Qu, "An improvement of d* algorithm for mobile robot path planning in partial unknown environment," in *2009 Second International Conference on Intelligent Computation Technology and Automation*, vol. 3, 2009, pp. 394–397.

[34] J. Chen, Y. Zhao, and X. Xu, "Improved rrt-connect based path planning algorithm for mobile robots," *IEEE Access*, vol. 9, pp. 145 988–145 999, 2021.

[35] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 1991, pp. 1012–1017 vol.2.

[36] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[37] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 601–606.

[38] R. Takemura and G. Ishigami, "Traversability-based rrt¡sup¿*¡/sup¿ for planetary rover path planning in rough terrain with lidar point cloud data," *Journal of Robotics and Mechatronics*, vol. 29, no. 5, pp. 838–846, 2017.

[39] M. Bekker, *Introduction to Terrain-vehicle Systems*. University of Michigan Press, 1969. [Online]. Available: https://books.google.it/books?id=Rn5TAAAAMAAJ

[40] L. Ding, Z. Deng, H. Gao, K. Nagatani, and K. Yoshida, "Planetary rovers' wheel–soil interaction mechanics: new challenges and applications for wheeled mobile robots," *Intelligent Service Robotics*, vol. 4, no. 1, pp. 17–38, Jan 2011. [Online]. Available: https://doi.org/10.1007/s11370-010-0080-5

[41] K. Yoshida, H. Hamano, and T. Watanabe, "Slip-based traction control of a planetary rover," in *Experimental Robotics VIII*, B. Siciliano and P. Dario, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 644–653.

[42] R. Gonzalez and K. Iagnemma, "Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges," *Journal of Field Robotics*, vol. 35, no. 4, pp. 564–577, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21761

[43] T. Naiki and T. Kubota, "1a2-c02 the evaluation of planetary rover for rough terrain," *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, vol. 2010, pp. _1A2–C02_1, 01 2010.

[44] D. R, B. Dharrun, P. G. Raj, C. Gowtham, and A. Kabilan, "Unmanned multipurpose all terrain rover using rocker bogie mechanism," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 1879–1882.

[45] T. Kubota and T. Naiki, "Novel mobility system with active suspension for planetary surface exploration," in *2011 Aerospace Conference*, 2011, pp. 1–9.

[46] H. Inotsume, S. Moreland, K. Skonieczny, and D. Wettergreen, "Parametric study and design guidelines for rigid wheels for planetary rovers," *Journal of Terramechanics*, vol. 85, pp. 39–57, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022489818301368

[47] M. Sutoh, K. Nagaoka, K. Nagatani, and K. Yoshida, "Design of wheels with grousers for planetary rovers traveling over loose soil," *Journal of Terramechanics*, vol. 50, no. 5, pp. 345–353, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022489813000499

[48] S. Haggart and J. Waydo, "The mobility system wheel design for nasa's mars science laboratory mission," in *11th European Conference of the International Society for Terrain-Vehicle Systems*, Torino, Italy, November 2008.

[49] H. J. Eisen, L. C. Wen, G. Hickey, and D. F. Braun, "Sojourner mars rover thermal performance," *SAE Transactions*, vol. 107, pp. 697–707, 1998. [Online]. Available: http://www.jstor.org/stable/44735797

[50] M. T. Lemmon, M. J. Wolff, M. D. Smith, R. T. Clancy, D. Banfield, G. A. Landis, A. Ghosh, P. H. Smith, N. Spanovich, B. Whitney, P. Whelley, R. Greeley, S. Thompson, J. F. Bell, and S. W. Squyres, "Atmospheric imaging results from the mars exploration rovers: Spirit and opportunity," *Science*, vol. 306, no. 5702, pp. 1753–1756, 2004. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1104474

[51] R. Welch, D. Limonadi, and R. Manning, "Systems engineering the curiosity rover: A retrospective," in *2013 8th International Conference on System of Systems Engineering*, 2013, pp. 70–75.

[52] J. Balaram, M. Aung, and M. P. Golombek, "The ingenuity helicopter on the perseverance rover," *Space Science Reviews*, vol. 217, no. 4, p. 56, May 2021. [Online]. Available: https://doi.org/10.1007/s11214-021-00815-w

[53] L. Ding, R. Zhou, Y. Yuan, H. Yang, J. Li, T. Yu, C. Liu, J. Wang, S. Li, H. Gao, Z. Deng, N. Li, Z. Wang, Z. Gong, G. Liu, J. Xie, S. Wang, Z. Rong, D. Deng, X. Wang, S. Han, W. Wan, L. Richter, L. Huang, S. Gou, Z. Liu, H. Yu, Y. Jia, B. Chen, Z. Dang, K. Zhang, L. Li, X. He, S. Liu, and K. Di, "A 2-year locomotive exploration and scientific investigation of the lunar farside by the yutu-2 rover," *Science Robotics*, vol. 7, no. 62, p. eabj6660, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abj6660

[54] J. Lai, Y. Xu, R. Bugiolacchi, X. Meng, L. Xiao, M. Xie, B. Liu, K. Di, X. Zhang, B. Zhou, S. Shen, and L. Xu, "First look by the yutu-2 rover at the deep subsurface structure at the lunar farside," *Nature Communications*, vol. 11, no. 1, p. 3426, Jul 2020. [Online]. Available: https://doi.org/10.1038/s41467-020-17262-w

[55] T. Zhang, S. Peng, Y. Jia, H. Tian, J. Sun, and C. Yan, "Slip estimation for mars rover zhurong based on data drive," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1676

[56] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev, "Rocky 7: A next generation mars rover prototype," *Advanced Robotics*, vol. 11, no. 4, p. 341 – 358, 1996, cited by: 102. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0031378529&doi=10.1163%2f156855397X00362&partnerID=40&md5=9118b39f331198f8198502072f24c065

[57] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Büttner, A. Dömel, W. Friedl, I. Grixa, M. Hellerer, H. Hirschmüller, M. Kassecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, "The lru rover for autonomous planetary exploration and its success in the spacebotcamp challenge," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2016, pp. 7–14.

[58] F. Cordes and A. Babu, "Sherpatt: A versatile hybrid wheeled-leg rover," in *13th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2016)*, 06 2016.

[59] F. Cordes, D. Kuehn, C. Oekermann, A. Babu, T. Stark, and F. Kirchner, "An active suspension system for a planetary rover," in *In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014)*, 06 2014.

[60] F. Cordes, F. Kirchner, and A. Babu, "Design and field testing of a rover with an actively articulated suspension system in a mars analog terrain," *Journal of Field Robotics*, vol. 35, no. 7, pp. 1149–1181, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21808

[61] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, "Modular reconfigurable robots in space applications," *Autonomous Robots*, vol. 14, no. 2, pp. 225–237, Mar 2003.

[62] T. de J. Mateo Sanguino, "50 years of rovers for planetary exploration: A retrospective review for future directions," *Robotics and Autonomous Systems*, vol. 94, pp. 172–185, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889016306030

[63] M. Massari, P. Massioni, S. Nebuloni, G. Sangiovanni, and F. Bernelli-Zazzera, "Realization and control of a prototype of legged rover for planetary exploration," in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 2005, pp. 863–868.

[64] H. Kolvenbach, P. Arm, E. Hampp, A. Dietsche, V. Bickel, B. Sun, C. Meyer, and M. Hutter, "Traversing steep and granular martian analog slopes with a dynamic quadrupedal robot," *Field Robotics*, vol. 2, pp. 910 – 939, 2022-05-23.

[65] M. Merz, A. A. Transeth, and G. Johansen, "Snake robots for space applications," 2017.

[66] R. Lin, W. Guo, and M. Li, "Novel design of legged mobile landers With decoupled landing and walking functions containing a rhombus joint," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, 08 2018, 061017. [Online]. Available: https://doi.org/10.1115/1.4040884

[67] R. Lin and W. Guo, "Creative design of legged mobile landers with multi-loop chains based on truss-mechanism transformation method," *Journal of Mechanisms and Robotics*, vol. 13, no. 1, 10 2020, 011013. [Online]. Available: https://doi.org/10.1115/1.4048256

[68] H. Kolvenbach, P. Arm, E. Hampp, A. Dietsche, V. Bickel, B. Sun, C. Meyer, and M. Hutter, "Traversing steep and granular martian analog slopes with a dynamic quadrupedal robot," 06 2021.

[69] D. Rodríguez-Martínez, M. Van Winnendael, and K. Yoshida, "High-speed mobility on planetary surfaces: A technical review," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1436–1455, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21912

[70] J. Gancet, A. Weiss, M. Tscheligi, M. Ilzkovitz, and R. Aked, "Human-robot collaboration in a planetary settlement setup : Collaborative work , interactions design and human factors," 2008.

[71] M. Arnhof, "Design of a human settlement on mars using in-situ resources," in *International Conference on Environmental Systems*, 2016.

[72] S. Pfeiffer, "The vision of "industrie 4.0" in the making—a case of future told, tamed, and traded," *NanoEthics*, vol. 11, no. 1, pp. 107–121, Apr 2017. [Online]. Available: https://doi.org/10.1007/s11569-016-0280-3

[73] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0 – a glimpse," *Procedia Manufacturing*, vol. 20, pp. 233–238, 2018, 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918300672

[74] M. Sony, "Pros and cons of implementing industry 4.0 for the organizations: a review and synthesis of evidence," *Production & Manufacturing Research*, vol. 8, no. 1, pp. 244–272, 2020. [Online]. Available: https://doi.org/10.1080/21693277.2020.1781705

[75] M. Mohamed, "Challenges and benefits of industry 4.0: an overview," *International Journal of Supply and Operations Management*, vol. 5, no. 3, pp. 256–265, 2018. [Online]. Available: http://www.ijsom.com/article_2767.html

[76] P. Osterrieder, L. Budde, and T. Friedli, "The smart factory as a key construct of industry 4.0: A systematic literature review," *International Journal of Production Economics*, vol. 221, p. 107476, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925527319302865

[77] F. Walas Mateo and A. Redchuk, "Iiot/iot and artificial intelligence/machine learning as a process optimization driver under industry 4.0 model iiot/iot e inteligencia artificial/aprendizaje automático como motor de optimización de procesos en el modelo de industria 4.0," *Journal of Computer Science and Technology*, vol. 21, 10 2021.

[78] J. Xu, M. Kovatsch, D. Mattern, F. Mazza, M. Harasic, A. Paschke, and S. Lucia, "A review on ai for smart manufacturing: Deep learning challenges

and solutions," *Applied Sciences*, vol. 12, no. 16, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/16/8239

[79] S. Harapanahalli, N. O. Mahony, G. V. Hernandez, S. Campbell, D. Riordan, and J. Walsh, "Autonomous navigation of mobile robots in factory environment," *Procedia Manufacturing*, vol. 38, pp. 1524–1531, 2019, 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing.

[80] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of Operations Research*, vol. 308, no. 1, pp. 125–143, Jan 2022.

[81] R. J. F. Barros, J. L. P. Silva Filho, J. V. S. Neto, and T. P. Nascimento, "An open-design warehouse mobile robot," in *2020 Latin American Robotics Symposium, 2020 Brazilian Symposium on Robotics and 2020 Workshop on Robotics in Education, LARS-SBR-WRE 2020*, 2020.

[82] H. Unger, T. Markert, and E. Müller, "Evaluation of use cases of autonomous mobile robots in factory environments," *Procedia Manufacturing*, vol. 17, pp. 254–261, 2018, 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USAGlobal Integration of Intelligent Manufacturing and Smart Industry for Good of Humanity. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918311600

[83] A. Markis, M. Papa, D. Kaselautzke, M. Rathmair, V. Sattinger, and M. Brandstötter, "Safety of mobile robot systems in industrial applications," in *Proceedings of the Joint ARW & OAGM Workshop 2019*, 2019.

[84] A. Souto, P. A. Prates, A. Lourenço, M. S. Al Maamari, F. Marques, D. Taranta, L. DoÓ, R. Mendonça, and J. Barata, "Fleet management system for autonomous mobile robots in secure shop-floor environments," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021, pp. 1–6.

[85] R. Hercik, R. Byrtus, R. Jaros, and J. Koziorek, "Implementation of autonomous mobile robot in smartfactory," *Applied Sciences*, vol. 12, no. 17, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/17/8912

[86] S. Bøgh, P. Gjøl Jensen, M. Kristjansen, K. Guldstrand Larsen, and U. Nyman, "Distributed fleet management in noisy environments via model-predictive control," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 32, no. 1, pp. 565–573, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/ICAPS/article/view/19843

[87] W. Blajer, W. Schiehlen, and W. Schirm, "Dynamic analysis of constrained multibody systems using inverse kinematics," *Mechanism and Machine Theory*, vol. 28, no. 3, p. 397 – 405, 1993, cited by: 12. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027591483&doi=10.1016%2f0094-114X%2893%2990078-A&partnerID=40&md5=3003c8b2642ced2e84eac38a2a03e93e

[88] P. Flores, J. Claro, J. Ambrósio, and H. Lankarani, "Multibody systems formulation," *Lecture Notes in Applied and Computational Mechanics*, vol. 34, pp. 23–45, 01 2008.

[89] P. E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems.* USA: Prentice-Hall, Inc., 1988.

[90] H. Kanchwala and A. Chatterjee, "Adams model validation for an all-terrain vehicle using test track data," *Advances in Mechanical Engineering*, vol. 11, no. 7, p. 1687814019859784, 2019. [Online]. Available: https://doi.org/10.1177/1687814019859784

[91] G. Wang and C. Xie, "Simulation analysis on ride comfort of hybrid heavy truck based on adams," in *Journal of Physics: Conference Series*, vol. 1865, no. 4, 2021, Conference paper, cited by: 1; All Open Access, Bronze Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85104604773&doi=10.1088%2f1742-6596%2f1865%2f4%2f042128&partnerID=40&md5=93a15c5304d684e9f34a8b8002f854e7

[92] H. Kanchwala and A. Chatterjee, "Adams model validation for an all-terrain vehicle using test track data," *Advances in Mechanical Engineering*, vol. 11, no. 7, p. 1687814019859784, 2019. [Online]. Available: https://doi.org/10.1177/1687814019859784

[93] N. Sesto Gorella, M. Caruso, P. Gallina, and S. Seriani, "Dynamically balanced pointing system for cubesats: Study and 3d printing manufacturing," *Robotics*, vol. 10, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/4/121

[94] Ratiu, Mariana, Rus, Alexandru, and Loredana Balas, Monica, "Modeling in adams of a 6r industrial robot," *MATEC Web Conf.*, vol. 184, p. 02006, 2018. [Online]. Available: https://doi.org/10.1051/matecconf/201818402006

[95] Y. J. Li and Z. W. Wu, "Modeling and simulation of a six-leg-wheel hybrid mobile robot based on adams," in *Advanced Mechanical Engineering*, ser. Applied Mechanics and Materials, vol. 26. Trans Tech Publications Ltd, 8 2010, pp. 194–197.

[96] L. Angel, C. Hernandez, and C. Diaz-Quintero, "Modeling, simulation and control of a differential steering type mobile robot," in *Chinese Control Conference, CCC*, 2013, p. 8757 – 8762. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84890446862&partnerID=40&md5=597c83eab8653db714249620cdc078b5

[97] X. Ning, C. Zhao, and J. Shen, "Dynamic analysis of car suspension using adams/car for development of a software interface for optimization," *Procedia Engineering*, vol. 16, pp. 333–341, 2011, international Workshop on Automobile, Power and Energy Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705811025938

[98] G. Aravind, S. Vishnu, K. Amarnath, U. Hithesh, P. Harikrishnan, P. Sreedharan, and G. Udupa, "Design, analysis and stability testing of lunar lander for soft-landing," *Materials Today: Proceedings*, vol. 24, pp. 1235 – 1243, 2020, int. Conf.

on Advances in Materials and Manufacturing Applications, IConAMMA 2018, 16th -18th August, 2018, India.

[99] G. Udupa, G. Sundaram, P. Poduval, A. A.P., K. Pillai, N. Kumar, N. Shaji, N. P.C., P. Ramacharan, and S. S. Rajan, "Certain investigations on soft lander for lunar exploration," *Procedia Computer Science*, vol. 133, pp. 393 – 400, 2018, int. Conf. on Robotics and Smart Manufacturing (RoSMa2018). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918309931

[100] V. Pham, J. Zhao, N. Goo, J. Lim, D. Hwang, and J. Park, "Landing stability simulation of a 1/6 lunar module with aluminum honeycomb dampers," *Int. Journal of Aeronautical and Space Sciences*, vol. 14, pp. 356–368, 2013.

[101] Yi Sun, Yabing Hu, R. Liu, and Z. Den, "Touchdown dynamic modeling and simulation of lunar lander," in *3rd Int. Symposium on Systems and Control in Aeronautics and Astronautics*, 2010, pp. 1320–1324.

[102] Y. Liu, S. Song, M. Li, and C. Wang, "Landing stability analysis for lunar landers using computer simulation experiments," *Int. Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417748441, 2017.

[103] M. Bonilla, E. Farnioli, C. Piazza, M. Catalano, G. Grioli, M. Garabini, M. Gabiccini, and A. Bicchi, "Grasping with soft hands," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 581–587.

[104] D. Song, L. Zheng, L. Wang, W. Qi, and Y. Li, "The co-simulation of humanoid robot based on solidworks, adams and simulink," in *Progress in Robotics*, J.-H. Kim, S. S. Ge, P. Vadakkepat, N. Jesse, A. Al Manum, S. Puthusserypady K, U. Rückert, J. Sitte, U. Witkowski, R. Nakatsu, T. Braunl, J. Baltes, J. Anderson, C.-C. Wong, I. Verner, and D. Ahlgren, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 10–18.

[105] D. Curry and Y. Deng, "Optimizing heavy equipment for handling bulk materials with adams-edem co-simulation," in *Springer Proceedings in Physics*, vol. 188, 2017, Conference paper, p. 1219 – 1224, cited by: 6. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85007364706&doi=10.1007%2f978-981-10-1926-5_126&partnerID=40&md5=28c7ce80bc4003490a556fe8a0eb6bc5

[106] X. Wang, S. Yang, W. Li, and Y. Wang, "Vibratory finishing co-simulation based on adams-edem with experimental validation," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 1, pp. 1175–1185, Apr 2018. [Online]. Available: https://doi.org/10.1007/s00170-018-1639-0

[107] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[108] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.

[109] V. B. Theja, P. Yachameni, S. Shakeera, and H. Venkataraman, "Integration of gazebo and ros for underwater vehicle environment," in *OCEANS 2022 - Chennai*, 2022, pp. 1–6.

[110] S. Zhou, B. Li, C. Ding, L. Lu, and C. Ding, "An efficient deep reinforcement learning framework for uavs," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*, 2020, pp. 323–328.

[111] C. Bernardeschi, A. Fagiolini, M. Palmieri, G. Scrima, and F. Sofia, "Ros/gazebo based simulation of co-operative uavs," in *Modelling and Simulation for Autonomous Systems*, J. Mazal, Ed. Cham: Springer International Publishing, 2019, pp. 321–334.

[112] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *Simulation, Modeling, and Programming for Autonomous Robots*, I. Noda, N. Ando, D. Brugali, and J. J. Kuffner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 400–411.

[113] M. Zhang, H. Qin, M. Lan, J. Lin, S. Wang, K. Liu, F. Lin, and B. M. Chen, "A high fidelity simulator for a quadrotor uav using ros and gazebo," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 002 846–002 851.

[114] F. D'Urso, C. Santoro, and F. F. Santoro, "An integrated framework for the realistic simulation of multi-uav applications," *Computers & Electrical Engineering*, vol. 74, pp. 196–209, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790618329161

[115] M. Sokolov, I. Afanasyev, R. Lavrenov, A. Sagitov, L. Sabirova, and E. Magid, "Modelling a crawler-type ugv for urban search and rescue in gazebo environment," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 22, pp. 360–363, 01 2017.

[116] A. Gallina, A. Gibbesch, R. Krenn, T. Uhl, and B. Schäfer, "Multibody simulation of planetary rover mobility in condition of uncertain soft terrain," *Procedia IUTAM*, vol. 13, pp. 118–126, 2015, dynamical Analysis of Multibody Systems with Design Uncertainties. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210983815000103

[117] K. Kenneth and M. Shillor, "Dynamic contact with signorini's condition and slip rate dependent friction," *Electronic Journal of Differential Equations*, vol. 2004, 06 2004.

[118] P. Flores and H. M. Lankarani, *Contact Force Models for Multibody Dynamics*. Cham: Springer International Publishing, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-30897-5

[119] P. Flores, M. Machado, M. Silva, and J. Martins, "On the continuous contact force models for soft materials in multibody dynamics," *Multibody System Dynamics*, vol. 25, pp. 357–375, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01333699

[120] L. Skrinjar, J. Slavič, and M. Boltežar, "A review of continuous contact-force models in multibody dynamics," *International Journal of Mechanical Sciences*, vol. 145, pp. 171–187, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020740317333611

[121] H. Hertz, "On the contact of rigid elastic solids and on hardness," *Ch 6: Assorted Papers*, 1882.

[122] H. Lankarani and P. Nikravesh, "Continuous contact force models for impact analysis in multibody systems," *Nonlinear Dynamics*, vol. 5, pp. 193–207, 03 1994.

[123] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, 01 2009.

[124] J. M. O'Kane, *A Gentle Introduction to ROS*. Independently published, Oct. 2013.

[125] Y. Pyo, H. Cho, L. Jung, and D. Lim, *ROS Robot Programming (English)*. ROBOTIS, 12 2017. [Online]. Available: http://community.robotsource.org/t/download-the-ros-robot-programming-book-for-free/51

[126] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ros," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017, pp. 1–5.

[127] Y. Du, C. Ai, and Z. Feng, "Research on navigation system of amr based on ros," in *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2020, pp. 117–121.

[128] S. Kuzin and G. Sziebig, "Sros: Educational, low-cost autonomous mobile robot design based on ros," in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 1052–1057.

[129] G. Karalekas, S. Vologiannidis, and J. Kalomiros, "Europa — a ros-based open platform for educational robotics," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, 2019, pp. 452–457.

[130] R. Mishra and A. Javed, "Ros based service robot platform," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018, pp. 55–59.

[131] L. Su, G. Qiu, W. Tang, and M. Chen, "A ros based open source simulation environment for robotics beginners," in *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*, 2021, pp. 286–291.

[132] G. Beraldo, N. Castaman, R. Bortoletto, E. Pagello, J. del R. Millán, L. Tonin, and E. Menegatti, "Ros-health: An open-source framework for neurorobotics," in *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2018, pp. 174–179.

[133] M. Marian, F. Stîngă, M.-T. Georgescu, H. Roibu, D. Popescu, and F. Manta, "A ros-based control application for a robotic platform using the gazebo 3d simulator," in *2020 21th International Carpathian Control Conference (ICCC)*, 2020, pp. 1–5.

[134] W. Yao, W. Dai, J. Xiao, H. Lu, and Z. Zheng, "A simulation system based on ros and gazebo for robocup middle size league," in *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, 2015, p. 54 – 59.

[135] S.-P. Chang, J.-I. Park, and K.-C. Lee, "Nonlinear dynamic analysis of spatially suspended elastic catenary cable with finite element method," *KSCE Journal of Civil Engineering*, vol. 12, pp. 121–128, 03 2008.

[136] Y. Su, Y. Qiu, P. Liu, J. Tian, Q. Wang, and X. Wang, "Dynamic modeling, workspace analysis and multi-objective structural optimization of the large-span high-speed cable-driven parallel camera robot," *Machines*, vol. 10, no. 7, 2022. [Online]. Available: https://www.mdpi.com/2075-1702/10/7/565

[137] D. S. D'Antonio, G. A. Cardona, and D. Saldana, "The catenary robot: Design and control of a cable propelled by two quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, p. 3857 – 3863, 2021, all Open Access, Green Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101810024&doi=10.1109%2fLRA.2021.3062603&partnerID=40&md5=90c70f518b3ee999310dca15e5ab1134

[138] C. M. Low, E. Yin-Kwee Ng, S. Narasimalu, K. H. Chua, and Y. Kim, "Adaptive discretisation and dual-rate time-stepping of mooring cable dynamics," *Ocean Engineering*, vol. 188, p. 106275, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801819304500

[139] M. Hall and A. Goupee, "Validation of a lumped-mass mooring line model with deepcwind semisubmersible model test data," *Ocean Engineering*, vol. 104, pp. 590–603, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801815002279

[140] X. Chen, B. Liu, and G. Le, "Numerical simulation research on the anchor last deployment of marine submersible buoy system based on vof method," *Journal of Marine Science and Engineering*, vol. 10, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2077-1312/10/11/1681

[141] X. Chen, B. Hong, Z. Lin, J. Hou, Z. Gao, and S. Lv, "Lumped mass model for flexible cable: A review," *Journal of Physics: Conference Series*, vol. 1995, no. 1, p. 012029, aug 2021. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1995/1/012029

[142] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric control of a quadrotor uav transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486–1498, Dec 2015. [Online]. Available: https://doi.org/10.1007/s12555-014-0304-0

[143] D. Lin, G. Mottola, M. Carricato, and X. Jiang, "Modeling and control of a cable-suspended sling-like parallel robot for throwing operations," *Applied Sciences*, vol. 10, no. 24, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/24/9067

[144] W. Rogers and M. S. C. (U.S.), *Apollo Experience Report: Lunar Module Landing Gear Subsystem*, ser. Apollo Experience Report: Lunar Module Landing Gear Subsystem. National Aeronautics and Space Administration, 1972, no. v. 6850. [Online]. Available: https://books.google.de/books?id=LIjG6Q3Wv14C

[145] W. J. O'Neil, R. P. Rudd, D. L. Farless, C. E. Hildebrand, R. T. Mitchell, K. H. Rourke, and E. A. Euler, "Viking navigation," NASA, JPL, California Institute of Technology Pasadena,California, Tech. Rep. NASA-CR-162917, JPL-PUB-78-38, November 1979.

[146] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt, "Mars pathfinder airbag impact attenuation system," American Institute of Aeronautics and Astronautics, East Lansing, Michigan, Tech. Rep. AIAA-95-1552-CP, January 2006.

[147] P. Cao, X. Hou, P. Xue, T. Tang, and Z. Deng, "Research for a modeling method of mars flexible airbag based on discrete element theory," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2017, pp. 351–356.

[148] A. Accomazzo, S. Lodiot, and V. Companys, "Rosetta mission operations for landing," *Acta Astronautica*, vol. 125, pp. 30 – 40, 2016, rosetta and Philae at comet 67P/Churyumov-Gerasimenko. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576516000230

[149] D. Woods, "Review of the soviet lunar exploration programme," *Spaceflight*, vol. 18, no. 7-8, pp. 273–290, 1976.

[150] V. Efanov and V. Dolgopolov, "The moon: from research to exploration (to the 50th anniversary of luna-9 and luna-10 spacecraft)," *Solar System Research*, vol. 51, no. 7, pp. 573–578, 2017.

[151] M. Malenkov, "Self-propelled automatic chassis of lunokhod-1: history of creation in episodes," *Frontiers of Mechanical Engineering*, vol. 11, no. 1, pp. 60–86, 2016.

[152] S. Li, X. Jiang, and T. Tao, "Guidance summary and assessment of the chang'e-3 powered descent and landing," *Journal of Spacecraft and Rockets*, vol. 53, 11 2015.

[153] W. Wu, D. Yu, C. Wang, J. Liu, Y. Tang, H. Zhang, and Z. Zhang, "Technological breakthroughs and scientific progress of the chang'e 4 mission," *Science China Information Sciences*, vol. 63, no. 10, 2020.

[154] C. Wang, H. Nie, J. Chen, and H. P. Lee, "The design and dynamic analysis of a lunar lander with semi-active control," *Acta Astronautica*, vol. 157, pp. 145 – 156, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576518315601

[155] J. Zhou, S. Jia, J. Qian, M. Chen, and J. Chen, "Improving the buffer energy absorption characteristics of movable lander-numerical and experimental studies," *Materials*, vol. 13, no. 15, 2020. [Online]. Available: https://www.mdpi.com/1996-1944/13/15/3340

[156] Y. Liu, S. Song, and C. Wang, "Multi-objective optimization on the shock absorber design for the lunar probe using nondominated sorting genetic algorithm ii," *Int. Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 1729881417720467, 2017.

[157] J. Grotzinger, J. Crisp, A. Vasavada, R. Anderson, C. Baker, R. Barry, D. Blake, P. Conrad, K. Edgett, B. Ferdowski, R. Gellert, J. Gilbert, M. Golombek, J. Gómez-Elvira, D. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. Malin, I. Mitrofanov, J. Simmonds, D. Vaniman, R. Welch, and R. Wiens, "Mars science laboratory mission and science investigation," *Space Science Reviews*, vol. 170, no. 1-4, pp. 5–56, 2012.

[158] Z. Yue, K. Di, Z. Liu, G. Michael, M. Jia, X. Xin, B. Liu, M. Peng, and J. Liu, "Lunar regolith thickness deduced from concentric craters in the ce-5 landing area," *Icarus*, vol. 329, pp. 46–54, 2019.

[159] K. Donaldson Hanna, D. Schrader, E. Cloutis, G. Cody, A. King, T. McCoy, D. Applin, J. Mann, N. Bowles, J. Brucato, J. Connolly, H.C., E. Dotto, L. Keller, L. Lim, B. Clark, V. Hamilton, C. Lantz, D. Lauretta, S. Russell, and P. Schofield, "Spectral characterization of analog samples in anticipation of osiris-rex's arrival at bennu: A blind test study," *Icarus*, vol. 319, pp. 701–723, 2019.

[160] E. Slyuta, "Physical and mechanical properties of the lunar soil (a review)," *Solar System Research*, vol. 48, pp. 330–353, 09 2014.

[161] L. Xu, H. Nie, J. Wan, Q. Lin, and J. Chen, "Analysis of landing impact performance for lunar lander based on flexible body," in *IEEE Int. Conf. on Computer Science and Automation Engineering*, vol. 1, 2011, pp. 10–13.

[162] J. Wan, H. Nie, J. Chen, and Q. Lin, "Modeling and simulation of lunar lander soft-landing using transient dynamics approach," in *2010 Int. Conf. on Computational and Information Sciences*, 2010, pp. 741–744.

[163] G. Zheng, H. Nie, J. Chen, C. Chen, and H. P. Lee, "Dynamic analysis of lunar lander during soft landing using explicit finite element method," *Acta Astronautica*, vol. 148, pp. 69 – 81, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576517311153

[164] D. Liang, H. Chai, and T. Chen, "Landing dynamic analysis for landing leg of lunar lander using abaqus/explicit," in *Proceedings of 2011 Int. Conf. on Electronic Mechanical Engineering and Information Technology*, vol. 8, 2011, pp. 4364–4367.

[165] S. Seriani, "A new mechanism for soft landing in robotic space exploration," *Robotics*, 12 2019.

[166] S. Seriani and P. Gallina, "Variable radius drum mechanisms," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, 11 2015, 021016.

[167] G. Endo, H. Yamada, A. Yajima, M. Ogata, and S. Hirose, "A passive weight compensation mechanism with a non-circular pulley and a spring," in *2010 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2010, pp. 3843–3848.

[168] L. Scalera, P. Gallina, S. Seriani, and A. Gasparetto, "Cable-based robotic crane (cbrc): design and implementation of overhead traveling cranes based on variable radius drums," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 474–485, 2018.

[169] D. Fedorov and L. Birglen, "Differential noncircular pulleys for cable robots and static balancing," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, 09 2018, 061001. [Online]. Available: https://doi.org/10.1115/1.4041213

[170] J.-w. Suh and K.-y. Kim, "Harmonious cable actuation mechanism for soft robot joints using a pair of noncircular pulleys," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, 2018.

[171] D. Fedorov and L. Birglen, "Design of a compliant mechanical device for upper leg rehabilitation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 870–877, 2019.

[172] L. Scalera, I. Palomba, E. Wehrle, A. Gasparetto, and R. Vidoni, "Natural motion for energy saving in robotic and mechatronic systems," *Applied Sciences*, vol. 9, no. 17, p. 3516, 2019.

[173] M. Li, W. Cheng, and R. Xie, "Design and experimental validation of two cam-based force regulation mechanisms," *Journal of Mechanisms and Robotics*, vol. 12, no. 3, 2020.

[174] W. D. Carrier, "Lunar soil grain size distribution," *The moon*, vol. 6, no. 3, pp. 250–263, Sep 1973. [Online]. Available: https://doi.org/10.1007/BF00562206

[175] J. Graf, *Lunar Soils Grain Size Catalog*, ser. NASA Reference Publication. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program, 1993. [Online]. Available: https://books.google.it/books?id=U446AQAAIAAJ

[176] M. Caruso, L. Scalera, P. Gallina, and S. Seriani, "Dynamic modeling and simulation of a robotic lander based on variable radius drums," *Applied Sciences*, vol. 10, no. 24, pp. 1–21, 2020.

[177] S. Seriani, L. Scalera, A. Gasparetto, and P. Gallina, "Preloaded structures for space exploration vehicles," in *IFToMM Symposium on Mechanism Design for Robotics*. Springer, 2018, pp. 129–137.

[178] S. Seriani, P. Gallina, L. Scalera, and V. Lughi, "Development of n-dof preloaded structures for impact mitigation in cobots," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, 2018.

[179] J. Grotzinger, J. Crisp, A. Vasavada, R. Anderson, C. Baker, R. Barry, D. Blake, P. Conrad, K. Edgett, B. Ferdowski, R. Gellert, J. Gilbert, M. Golombek, J. Gómez-Elvira, D. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. Malin, I. Mitrofanov, J. Simmonds, D. Vaniman, R. Welch, and R. Wiens, "Mars science laboratory mission and science investigation," *Space Science Reviews*, vol. 170, no. 1-4, pp. 5–56, 2012. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84866739490&doi=10.1007%2fs11214-012-9892-2&partnerID=40&md5=63ed823ef8b441dd5d9a900263515d0b

[180] P. Mahaffy, C. Webster, M. Cabane, P. Conrad, P. Coll, S. Atreya, R. Arvey, M. Barciniak, M. Benna, L. Bleacher, W. Brinckerhoff, J. Eigenbrode, D. Carignan, M. Cascia, R. Chalmers, J. Dworkin, T. Errigo, P. Everson,

H. Franz, R. Farley, S. Feng, G. Frazier, C. Freissinet, D. Glavin, D. Harpold, D. Hawk, V. Holmes, C. Johnson, A. Jones, P. Jordan, J. Kellogg, J. Lewis, E. Lyness, C. Malespin, D. Martin, J. Maurer, A. McAdam, D. McLennan, T. Nolan, M. Noriega, A. Pavlov, B. Prats, E. Raaen, O. Sheinman, D. Sheppard, J. Smith, J. Stern, F. Tan, M. Trainer, D. Ming, R. Morris, J. Jones, C. Gundersen, A. Steele, J. Wray, O. Botta, L. Leshin, T. Owen, S. Battel, B. Jakosky, H. Manning, S. Squyres, R. Navarro-González, C. McKay, F. Raulin, R. Sternberg, A. Buch, P. Sorensen, R. Kline-Schoder, D. Coscia, C. Szopa, S. Teinturier, C. Baffes, J. Feldman, G. Flesch, S. Forouhar, R. Garcia, D. Keymeulen, S. Woodward, B. Block, K. Arnett, R. Miller, C. Edmonson, S. Gorevan, and E. Mumm, "The sample analysis at mars investigation and instrument suite," *Space Science Reviews*, vol. 170, no. 1-4, pp. 401–478, 2012. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84866738553&doi=10.1007%2fs11214-012-9879-z&partnerID=40&md5=9c817998b815bf740f6ab90b83ca6c0a

[181] M. Lapotre, R. Ewing, M. Lamb, W. Fischer, J. Grotzinger, D. Rubin, K. Lewis, M. Ballard, M. Day, S. Gupta, S. Banham, N. Bridges, D. Marais, A. Fraeman, J. Grant, K. Herkenhoff, D. Ming, M. Mischna, M. Rice, D. Sumner, A. Vasavada, and R. Yingst, "Large wind ripples on mars: A record of atmospheric evolution," *Science*, vol. 353, no. 6294, pp. 55–58, 2016. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84976905111&doi=10.1126%2fscience.aaf3206&partnerID=40&md5=58a9a69a4dd117b98684196e463df21a

[182] C. Webster, P. Mahaffy, S. Atreya, J. Moores, G. Flesch, C. Malespin, C. McKay, G. Martinez, C. Smith, J. Martin-Torres, J. Gomez-Elvira, M.-P. Zorzano, M. Wong, M. Trainer, A. Steele, J. Archer, D., B. Sutter, P. Coll, C. Freissinet, P.-Y. Meslin, R. Gough, C. House, A. Pavlov, J. Eigenbrode, D. Glavin, J. Pearson, D. Keymeulen, L. Christensen, S. Schwenzer, R. Navarro-Gonzalez, J. Pla-García, S. Rafkin, Á. Vicente-Retortillo, H. Kahanpää, D. Viudez-Moreiras, M. Smith, A.-M. Harri, M. Genzer, D. Hassler, M. Lemmon, J. Crisp, S. Sander, R. Zurek, and A. Vasavada, "Background levels of methane in mars' atmosphere show strong seasonal variations," *Science*, vol. 360, no. 6393, pp. 1093–1096, 2018. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85048593070&doi=10.1126%2fscience.aaq0131&partnerID=40&md5=c5c47e521969c7a52de7d0c77f7918eb

[183] A. Wedler, M. Schuster, M. Müller, B. Vodermayer, L. Meyer, R. Giubilato, M. Vayugundla, M. Smisek, A. Dömel, F. Steidle, P. Lehner, S. Schröder, E. Staudinger, B. Foing, and J. Reill, "German aerospace center's advanced robotic technology for future lunar scientific missions: Dlr's advanced robotic technology," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2188, 2021. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85096630167&doi=10.1098%2frsta.2019.0574rsta20190574&partnerID=40&md5=ee7f9518a9b325072fb867f6236426aa

[184] L. Witte, A. Heffels, M. Killian, M. Knapmeyer, C. Lange, N. Toth, G. Tsakyridis, and A. Wedler, "A robotically deployable lunar surface science station and its validation in a moon-analogue environment," *Planetary and Space Science*, vol. 193, 2020. [Online]. Available: https:

//www.scopus.com/inward/record.uri?eid=2-s2.0-85090560160&doi=10.1016%2fj.pss.2020.105080&partnerID=40&md5=5275e75d8960e28d039a175d560037bc

[185] S. Campagnola, C. Yam, Y. Tsuda, O. Naoko, and Y. Kawakatsu, "Mission analysis for the martian moons explorer (mmx) mission," *Acta Astronautica*, vol. 146, pp. 409–417, 2018. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85044142646&doi=10.1016%2fj.actaastro.2018.03.024&partnerID=40&md5=ea4636fddbff7b082cfb9cf7f01a5533

[186] S. Seriani, P. Gallina, and A. Wedler, "A modular cable robot for inspection and light manipulation on celestial bodies," *Acta Astronautica*, vol. 123, pp. 145–153, 2016.

[187] M. Schuster, M. Muller, S. Brunner, H. Lehner, P. Lehner, R. Sakagami, A. Domel, L. Meyer, B. Vodermayer, R. Giubilato, M. Vayugundla, J. Reill, F. Steidle, I. Von Bargen, K. Bussmann, R. Belder, P. Lutz, W. Sturzl, M. Smisek, M. Moritz, S. Stoneman, A. Prince, B. Rebele, M. Durner, E. Staudinger, S. Zhang, R. Pohlmann, E. Bischoff, C. Braun, S. Schroder, E. Dietz, S. Frohmann, A. Borner, H.-W. Hubers, B. Foing, R. Triebel, A. Albu-Schaffer, and A. Wedler, "The arches space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5315–5322, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088702238&doi=10.1109%2fLRA.2020.3007468&partnerID=40&md5=5e8facead99b54a6f1a5e15ceef6dcbe

[188] B. Muirhead and A. Karp, "Mars sample return lander mission concepts," in *IEEE Aerospace Conference Proceedings*, vol. 2019-March, 2019. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068340074&doi=10.1109%2fAERO.2019.8742215&partnerID=40&md5=ebb71e06e6369ac20afdc34e03197710

[189] B. Muirhead, A. Nicholas, J. Umland, O. Sutherland, and S. Vijendran, "Mars sample return campaign concept status," *Acta Astronautica*, vol. 176, pp. 131–138, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85086799263&doi=10.1016%2fj.actaastro.2020.06.026&partnerID=40&md5=959cae9e5a91ece66f522ec2251f503e

[190] S. Mallapaty, "China has landed its first rover on mars - here's what happens next," *Nature*, vol. 593, no. 7859, pp. 323–324, 2021, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106182816&doi=10.1038%2fd41586-021-01301-7&partnerID=40&md5=1ad25ecce7ef9d2b78ae12980503f411

[191] B. Oglesby, A. Prince, G. Story, and A. Kam, "Qualification of a hybrid propulsion system for a mars ascent vehicle," in *IEEE Aerospace Conference Proceedings*, vol. 2019-March, 2019. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068340943&doi=10.1109%2fAERO.2019.8741599&partnerID=40&md5=7ea1d0ea8c3fa9a6ba2b23299d335bb7

[192] Y. Tsuda, T. Saiki, F. Terui, S. Nakazawa, M. Yoshikawa, S.-I. Watanabe, and H. P. Team, "Hayabusa2 mission status: Landing, roving and

cratering on asteroid ryugu," *Acta Astronautica*, vol. 171, pp. 42–54, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85080149939&doi=10.1016%2fj.actaastro.2020.02.035&partnerID=40&md5=a0435b701b23d61c2311f41ecbae3ae6

[193] M. Schuster, S. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, O. Porges, J. Reill, S. Riedel, M. Vayugundla, B. Vodermayer, T. Bodenmüller, C. Brand, W. Friedl, I. Grixa, H. Hirschmüller, M. Kaßecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, "Towards autonomous planetary exploration: The lightweight rover unit (lru), its success in the spacebotcamp challenge, and beyond," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 93, no. 3-4, pp. 461–494, 2019. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85033398832&doi=10.1007%2fs10846-017-0680-9&partnerID=40&md5=748202f6fa6c2074debf61954b00d941

[194] F. Vulpi, A. Milella, F. Cordes, R. Domínguez, and G. Reina, "Deep terrain estimation for planetary rovers," in *15th International Symposium on Artificial Intelligence, Robotics and Automation in Space, iSAIRAS-2020.*, October 2020.

[195] S. Seriani, P. Gallina, L. Scalera, and V. Lughi, "Development of n-DoF Preloaded Structures for Impact Mitigation in Cobots," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, 07 2018, 051009. [Online]. Available: https://doi.org/10.1115/1.4040632

[196] S. Seriani, L. Scalera, A. Gasparetto, and P. Gallina, "Preloaded structures for space exploration vehicles," in *Mechanism Design for Robotics*, A. Gasparetto and M. Ceccarelli, Eds. Cham: Springer International Publishing, 2019, pp. 129–137.

[197] G. Reina and M. Foglia, "On the mobility of all-terrain rovers," *Industrial Robot: An International Journal*, vol. 40, pp. 5–5, 03 2013.

[198] G. Reina, "Cross-coupled control for all-terrain rovers," *Sensors*, vol. 13, no. 1, pp. 785–800, 2013. [Online]. Available: https://www.mdpi.com/1424-8220/13/1/785

[199] T. Kane and D. Levinson, *Dynamics, Theory and Applications*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1985. [Online]. Available: https://books.google.it/books?id=g99oQgAACAAJ

[200] A. S. RAMBELY, N. A. HALIM, and R. R. AHMAD, "A numerical comparison of langrange and kane's methods of an arm segment," *International Journal of Modern Physics: Conference Series*, vol. 09, pp. 68–75, 2012. [Online]. Available: https://doi.org/10.1142/S2010194512005119

[201] R. S. Pal, "Modelling of helicopter underslung dynamics using kane's method," *IFAC-PapersOnLine*, vol. 53, no. 1, pp. 536–542, 2020, 6th Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320301099

[202] R. Lindemann, *Dynamic modeling and simulation of planetary rovers.* AIAA, 1992. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1992-1269

[203] F. Benamar and C. Grand, "Quasi-static motion simulation and slip prediction of articulated planetary rovers using a kinematic approach," *Journal of Mechanisms and Robotics*, vol. 5, no. 2, Mar 2013, 021002. [Online]. Available: https://doi.org/10.1115/1.4023873

[204] F. Chen and G. Genta, "Dynamic modeling of wheeled planetary rovers: A model based on the pseudo-coordiates approach," *Acta Astronautica*, vol. 81, no. 1, pp. 288–305, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576512002469

[205] Q. Fu and V. Krovi, "Articulated Wheeled Robots: Exploiting Reconfigurability and Redundancy," in *ASME 2008 Dynamic Systems and Control Conference, Parts A and B*, ser. Dynamic Systems and Control Conference, 10 2008, pp. 653–660. [Online]. Available: https://doi.org/10.1115/DSCC2008-2193

[206] J. Wong, *Theory of Ground Vehicles*. Wiley, 2001. [Online]. Available: https://books.google.it/books?id=LH8wd8im13AC

[207] K. Yoshida and H. Hamano, "Motion dynamics of a rover with slip-based traction model," *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, vol. 2002, no. 3, pp. 3155–3160, 2002.

[208] Y. chun Yang, J. song Bao, Y. Jin, and Y.-I. Cheng, "A virtual simulation environment for lunar rover: Framework and key technologies," *International Journal of Advanced Robotic Systems*, vol. 5, no. 2, p. 16, 2008. [Online]. Available: https://doi.org/10.5772/5651

[209] G. Ishigami, A. Miwa, K. Nagatani, and K. Yoshida, "Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil," *Journal of Field Robotics*, vol. 24, no. 3, pp. 233–250, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20187

[210] B. Schäfer, A. Gibbesch, R. Krenn, and B. Rebele, "Planetary rover mobility simulation on soft and uneven terrain," *Vehicle System Dynamics*, vol. 48, no. 1, pp. 149–169, 2010. [Online]. Available: https://doi.org/10.1080/00423110903243224

[211] A. Gibbesch and B. Schäfer, "Multibody system modelling and simulation of plenetary rover mobility on soft terrain," in *ISAIRAS 2005 Conference*, 08 2005.

[212] R. Krenn and G. Hirzinger, "Simulation of rover locomotion on sandy terrain – modeling verification and validation," in *10th ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA 2008*, 11 2008.

[213] M. Sutoh, "Traveling performance analysis of planetary rovers using a repeatable test system in vacuum," *Journal of Terramechanics*, vol. 95, pp. 15–24, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002248982100015X

[214] C. Chen, M. Shu, Y. Wang, L. Ding, H. Gao, H. Liu, and S. Zhou, "Simultaneous control of trajectory tracking and coordinated allocation of rocker-bogie planetary rovers," *Mechanical Systems and Signal Processing*, vol. 151, p. 107312, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327020306981

[215] H. M. Lankarani and P. E. Nikravesh, "A Contact Force Model With Hysteresis Damping for Impact Analysis of Multibody Systems," *Journal of Mechanical Design*, vol. 112, no. 3, pp. 369–376, 09 1990. [Online]. Available: https://doi.org/10.1115/1.2912617

[216] K. H. Hunt and F. R. E. Crossley, "Coefficient of Restitution Interpreted as Damping in Vibroimpact," *Journal of Applied Mechanics*, vol. 42, no. 2, pp. 440–445, 06 1975. [Online]. Available: https://doi.org/10.1115/1.3423596

[217] L. Skrinjar, J. Slavič, and M. Boltežar, "A review of continuous contact-force models in multibody dynamics," *International Journal of Mechanical Sciences*, vol. 145, pp. 171–187, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020740317333611

[218] S. Seriani, P. Gallina, and A. Wedler, "Dynamics of a tethered rover on rough terrain," *Mechanisms and Machine Science*, vol. 47, pp. 355–361, 2017, cited By 5. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84996928614&doi=10.1007%2f978-3-319-48375-7_38&partnerID=40&md5=a5324f31988cd771f0e672271d55d352

[219] M. Caruso, L. Bregant, P. Gallina, and S. Seriani, "Design and multibody dynamic analysis of the archimede space exploration rover," *Acta Astronautica*, vol. 194, pp. 229–241, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522000571

[220] C. Makkar, W. E. Dixon, W. G. Sawyer, and G. Hu, "A new continuously differentiable friction model for control systems design," in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 2005, pp. 600–605.

[221] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, "Chapter 2 - motion modeling for mobile robots," in *Wheeled Mobile Robotics*, G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, Eds. Butterworth-Heinemann, 2017, pp. 13–59. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128042045000020

[222] W. H. Baier, "Vehicle steering fundamentals," *SAE Transactions*, vol. 69, pp. 256–268, 1961. [Online]. Available: http://www.jstor.org/stable/44553923

[223] M. Veneri and M. Massaro, "The effect of ackermann steering on the performance of race cars," *Vehicle System Dynamics*, vol. 59, pp. 1–21, 02 2020.

[224] H. Hlavacs, "A 2d car physics model based on ackermann steering," in *The Fourth Annual International Conference in Game Design and Technology 2006 (GDTW 2006)*, 2006. [Online]. Available: http://eprints.cs.univie.ac.at/669/

[225] M. Topaç, M. Karaca, A. Başdemir, and B. Kuleli, "Buckling safety assessment for the multi-axle steering linkage of an 8x8 special purpose vehicle," *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, vol. 15, pp. 321–327, 12 2019.

[226] H. Chen, S. Chen, R. Zhou, X. Huang, and S. Zhu, "Research on four-wheel independent steering intelligent control strategy based on minimum load," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 9, p. e6145, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6145

[227] K. Spentzas, I. Alkhazali, and M. Demic, "Kinematics of four-wheel-steering vehicles," *Forschung im Ingenieurwesen*, vol. 66, pp. 211–216, 05 2001.

[228] W. Paszkowiak, T. Bartkowiak, and M. Pelic, "Kinematic model of a logistic train with a double ackermann steering system," *International Journal of Simulation Modelling*, 2021.

[229] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Büttner, A. Dömel, W. Friedl, I. Grixa, M. Hellerer, H. Hirschmüller, M. Kassecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, "The lru rover for autonomous planetary exploration and its success in the spacebotcamp challenge," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2016, pp. 7–14.

[230] M. Manz, R. Sonsalla, J. Hilljegerdes, C. Oekermann, J. Schwendner, S. Bartsch, and S. Ptacek, "Mechanical design of a rover for mobile manipulation in uneven terrain in the context of the spacebot cup," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, 06 2014.

[231] E. Tunstel, M. Maimone, A. Trebi-Ollennu, J. Yen, R. Petras, and R. Willson, "Mars exploration rover mobility and robotic arm operational performance," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, 2005, pp. 1807–1814 Vol. 2.

[232] R. C. Moeller, L. Jandura, K. Rosette, M. Robinson, J. Samuels, M. Silverman, K. Brown, E. Duffy, A. Yazzie, E. Jens, I. Brockie, L. White, Y. Goreva, T. Zorn, A. Okon, J. Lin, M. Frost, C. Collins, J. B. Williams, A. Steltzner, F. Chen, and J. Biesiadecki, "The sampling and caching subsystem (scs) for the scientific exploration of jezero crater by the mars 2020 perseverance rover," *Space Science Reviews*, vol. 217, no. 1, p. 5, Dec 2020. [Online]. Available: https://doi.org/10.1007/s11214-020-00783-7

[233] K. A. Farley, K. H. Williford, K. M. Stack, R. Bhartia, A. Chen, M. de la Torre, K. Hand, Y. Goreva, C. D. K. Herd, R. Hueso, Y. Liu, J. N. Maki, G. Martinez, R. C. Moeller, A. Nelessen, C. E. Newman, D. Nunes, A. Ponce, N. Spanovich, P. A. Willis, L. W. Beegle, J. F. Bell, A. J. Brown, S.-E. Hamran, J. A. Hurowitz, S. Maurice, D. A. Paige, J. A. Rodriguez-Manfredi, M. Schulte, and R. C. Wiens, "Mars 2020 mission overview," *Space Science Reviews*, vol. 216, no. 8, p. 142, Dec 2020. [Online]. Available: https://doi.org/10.1007/s11214-020-00762-y

[234] A. Muralidharan and Y. Mostofi, "Path planning for minimizing the expected cost until success," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 466–481, 2019.

[235] P. Abad-Manterola, I. A. D. Nesnas, and J. Burdick, "Motion planning on steep terrain for the tethered axel rover," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4188–4195.

[236] B. Donald, L. Gariepy, and D. Rus, "Experiments in constrained prehensile manipulation: Distributed manipulation with ropes," *Lecture Notes in Control and Information Sciences*, 10 1999.

[237] ——, "Distributed manipulation of multiple objects using ropes," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 450–457 vol.1.

[238] T. Maneewarn and P. Detudom, "Mechanics of cooperative nonprehensile pulling by multiple robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2004–2009.

[239] A. Yamashita, J. Sasaki, J. Ota, and T. Arai, "Cooperative manipulation of objects by multiple mobile robots with tools *," 1998.

[240] R. Soares, E. Bicho, T. Machado, and W. Erlhagen, "Object transportation by multiple mobile robots controlled by attractor dynamics: Theory and implementation," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10 2007, pp. 937 – 944.

[241] J. Sasaki, J. Ota, E. Yoshida, D. Kurabayashi, and T. Arai, "Cooperating grasping of a large object by multiple mobile robots," *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1205–1210 vol.1, 1995.

[242] A. Yufka, O. Parlaktuna, and M. Ozkan, "Formation-based cooperative transportation by a group of non-holonomic mobile robots," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 3300–3307.

[243] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and Y. Doroftei, "Cooperative lifting and transport by a group of mobile robots," in *DARS 2014*, 2014.

[244] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying mpc for nonprehensile object manipulation with a nonholonomic mobile robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 032–11 038.

[245] S. Krivic and J. Piater, "Pushing corridors for delivering unknown objects with a mobile robot," *Autonomous Robots*, vol. 43, no. 6, pp. 1435–1452, Aug 2019. [Online]. Available: https://doi.org/10.1007/s10514-018-9804-8

[246] P. Kolhe, N. Dantam, and M. Stilman, "Dynamic pushing strategies for dynamically stable mobile manipulators," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3745–3750.

[247] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Characterization of grasp configurations for multi-robot object pushing," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2021, pp. 38–46.

[248] I. Mas and C. A. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I*, 2012.

[249] F. Ohashi, K. Kaminishi, J. Figueroa, H. Kato, and J. Ota, "Transportation of a large object by small mobile robots with handcarts and outrigger," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014, pp. 70–75.

[250] F. Ohashi, K. Kaminishi, J. D. Figueroa Heredia, H. Kato, T. Ogata, T. Hara, and J. Ota, "Realization of heavy object transportation by mobile robots using handcarts and outrigger," *ROBOMECH Journal*, vol. 3, no. 1, p. 27, Nov 2016. [Online]. Available: https://doi.org/10.1186/s40648-016-0066-y

[251] S. Iglesias-Baniela, J. Vinagre-Ríos, and J. M. Pérez-Canosa, "Ship handling in unprotected waters: A review of new technologies in escort tugs to improve safety," *Applied Mechanics*, vol. 2, no. 1, pp. 46–62, 2021. [Online]. Available: https://www.mdpi.com/2673-3161/2/1/4

[252] G. Wu, X. Zhao, Y. Sun, and L. Wang, "Cooperative maneuvering mathematical modeling for multi-tugs towing a ship in the port environment," *Journal of Marine Science and Engineering*, vol. 9, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/4/384

[253] X. Chen, B. Hong, Z. Lin, J. Hou, Z. Gao, and S. Lv, "Lumped mass model for flexible cable: A review," in *Journal of Physics: Conference Series*, vol. 1995, no. 1, 2021. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85112750349&doi=10.1088

[254] J. Park and N. Kim, "Dynamics modeling of a semi-submersible autonomous underwater vehicle with a towfish towed by a cable," *International Journal of Naval Architecture and Ocean Engineering*, vol. 7, no. 2, pp. 409–425, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2092678216300899

[255] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 643–659.

[256] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor uavs with a suspended payload," in *Proceedings of the IEEE Conference on Decision and Control*, 12 2013, pp. 5510–5515.

[257] Z. Li, J. Erskine, S. Caro, and A. Chriette, "Design and control of a variable aerial cable towed system," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 636–643, 2020.

[258] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, Jan 2011. [Online]. Available: https://doi.org/10.1007/s10514-010-9205-0

[259] Q. Jiang and V. Kumar, "The inverse kinematics of 3-d towing," in *Advances in Robot Kinematics: Motion in Man and Machine*, J. Lenarcic and M. M. Stanisic, Eds. Dordrecht: Springer Netherlands, 2010, pp. 321–328.

[260] D. Ma, S. Wang, M. Yang, and Y. Dong, "Dynamic simulation of aerial towed decoy system based on tension recurrence algorithm," *Chinese Journal of Aeronautics*, vol. 29, no. 6, pp. 1484–1495, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936116301704

[261] P. Abad-Manterola, J. Edlund, J. Burdick, A. Wu, T. Oliver, I. A. Nesnas, and J. Cecava, "Axel," *IEEE Robotics and Automation Magazine*, vol. 16, no. 4, pp. 44–52, 2009.

[262] G. Paulsen, S. Farritor, T. Huntsberger, and H. Aghazarian, "All terrain exploration with the cliff-bot system," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 721–726.

[263] P. Cheng, J. Fink, V. Kumar, and J.-S. Pang, "Cooperative Towing With Multiple Robots," *Journal of Mechanisms and Robotics*, vol. 1, no. 1, 08 2008, 011008. [Online]. Available: https://doi.org/10.1115/1.2960539

[264] Y.-H. Kim and D. A. Shell, "Using a compliant, unactuated tail to manipulate objects," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 223–230, 2017.

[265] ——, "Bound to help: cooperative manipulation of objects via compliant, unactuated tails," *Autonomous Robots*, vol. 42, no. 8, pp. 1563–1582, Dec 2018. [Online]. Available: https://doi.org/10.1007/s10514-018-9718-5

[266] S. Wilson, A. Buffin, S. C. Pratt, and S. Berman, "Multi-robot replication of ant collective towing behaviours," *Royal Society open science*, vol. 5, no. 10, pp. 180 409–180 409, Oct 2018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/30473810

[267] M. Tarokh and G. McDermott, "Kinematics modeling and analyses of articulated rovers," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 539–553, 2005.

[268] Q. Lou, F. González, and J. Kövecses, "Kinematic modeling and state estimation of exploration rovers," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1311–1318, 2019.

[269] L. Ding, L. Huang, S. Li, H. Gao, H. Deng, Y. Li, and G. Liu, "Definition and application of variable resistance coefficient for wheeled mobile robots on deformable terrain," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 894–909, 2020.

[270] A. Azimi, J. Kövecses, and J. Angeles, "Wheel–soil interaction model for rover simulation and analysis using elastoplasticity theory," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1271–1288, 2013.

[271] T. P. Setterfield and A. Ellery, "Terrain response estimation using an instrumented rocker-bogie mobility system," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 172–188, 2013.

[272] L. Ojeda, D. Cruz, G. Reina, and J. Borenstein, "Current-based slippage detection and odometry correction for mobile robots and planetary rovers," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 366–378, 2006.

[273] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, "Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 921–927, 2004.

[274] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986. [Online]. Available: https://doi.org/10.1177/027836498600500303

[275] S. Goyal, A. Ruina, and J. Papadopoulos, "Limit surface and moment function descriptions of planar sliding," in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 794–799 vol.2.

[276] ——, "Planar sliding with dry friction part 1. limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0043164891901043

[277] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996. [Online]. Available: https://doi.org/10.1177/027836499601500603

[278] S. Takagi and Y. Okawa, "Rule-based control of a mobile robot for the push-a-box operation," in *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, 1991, pp. 1338–1343 vol.3.

[279] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 533–556, 1996. [Online]. Available: https://doi.org/10.1177/027836499601500602

[280] A. Zarei Khabjani, H. Karimpour, and M. Keshmiri, "Robotic box pushing under indeterminate anisotropic friction properties," *International Journal of Dynamics and Control*, vol. 9, no. 3, pp. 872–884, Sep 2021. [Online]. Available: https://doi.org/10.1007/s40435-020-00720-0

[281] I. Kao and M. R. Cutkosky, "Quasistatic manipulation with compliance and sliding," *The International Journal of Robotics Research*, vol. 11, no. 1, pp. 20–40, 1992. [Online]. Available: https://doi.org/10.1177/027836499201100102

[282] N. Xydas and I. Kao, "Modeling of contact mechanics with experimental results for soft fingers," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 1, 1998, pp. 488–493 vol.1.

[283] M. M. G. Ardakani, J. Bimbo, and D. Prattichizzo, "Quasi-static analysis of planar sliding using friction patches," *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1775–1795, 2020. [Online]. Available: https://doi.org/10.1177/0278364920929082

[284] A. Fakhari, M. Keshmiri, and M. Keshmiri, "Dynamic Modeling and Slippage Analysis in Object Manipulation by Soft Fingers," in *ASME International Mechanical Engineering Congress and Exposition*, vol. Volume 4A: Dynamics, Vibration, and Control, 11 2014, v04AT04A056. [Online]. Available: https://doi.org/10.1115/IMECE2014-38498

[285] A. Fakhari, M. Keshmiri, and I. Kao, "Development of realistic pressure distribution and friction limit surface for soft-finger contact interface of robotic hands," *J. Intell. Robotics Syst.*, vol. 82, no. 1, p. 39–50, Apr. 2016. [Online]. Available: https://doi.org/10.1007/s10846-015-0267-2

[286] A. Fakhari, I. Kao, and M. Keshmiri, "Modeling and control of planar slippage in object manipulation using robotic soft fingers," *ROBOMECH Journal*, vol. 6, no. 1, p. 15, Nov 2019. [Online]. Available: https://doi.org/10.1186/s40648-019-0143-0

[287] M. Caruso, L. Bregant, P. Gallina, and S. Seriani, "Design and multibody dynamic analysis of the archimede space exploration rover," *Acta Astronautica*, vol. 194, pp. 229–241, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522000571

[288] L. F. Shampine and M. W. Reichelt, "The matlab ode suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997. [Online]. Available: https://doi.org/10.1137/S1064827594276424

[289] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679.

[290] S. Ajwad and J. Iqbal, "Recent advances and applications of tethered robotic systems," *Science International*, vol. 26, pp. 2045–2051, 01 2014.

[291] X. Zhang and Q.-C. Pham, "Planning coordinated motions for tethered planar mobile robots," *Robotics and Autonomous Systems*, vol. 118, pp. 189 – 203, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018309710

[292] L. Fagiano, "Systems of tethered multicopters: Modeling and control design," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4610 – 4615, 2017, 20th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631731039X

[293] M. Tognon and A. Franchi, "Nonlinear observer for the control of bi-tethered multi aerial robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1852–1857.

[294] B. Kosarnovsky and S. Arogeti, "A string of tethered drones - system dynamics and control," in *2019 European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–6.

[295] T. W. McLain and S. M. Rock, "Experimental Measurement of ROV Tether Tension," in *Proceedings of Intervention/ROV '92*. San Diego, CA: IEEE, June 1992, pp. 291–296. [Online]. Available: http://www.stanford.edu/group/arl/cgi-bin/drupal/sites/default/files/public/publications/McLainR92.pdf

[296] P. Cheng, J. Fink, V. Kumar, and J.-S. Pang, "Cooperative Towing With Multiple Robots," *Journal of Mechanisms and Robotics*, vol. 1, no. 1, 08 2008, 011008. [Online]. Available: https://doi.org/10.1115/1.2960539

[297] Q. Jiang and V. Kumar, "Determination and stability analysis of equilibrium configurations of objects suspended from multiple aerial robots," *Journal of Mechanisms and Robotics*, vol. 4, no. 2, 2012. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84859325476&doi=10.1115%2f1.4005588&partnerID=40&md5=86eedefbad56be7b43a41c2e901fb896

[298] A. Mohiuddin, Y. Zweiri, R. Almadhoun, T. Taha, and D. Gan, "Energy distribution in Dual-UAV collaborative transportation through load sharing," *Journal of Mechanisms and Robotics*, pp. 1–14, 04 2020. [Online]. Available: https://doi.org/10.1115/1.4046912

[299] J. Erskine, A. Chriette, and S. Caro, "Wrench Analysis of Cable-Suspended Parallel Robots Actuated by Quadrotor Unmanned Aerial Vehicles," *Journal of Mechanisms and Robotics*, vol. 11, no. 2, 02 2019, 020909. [Online]. Available: https://doi.org/10.1115/1.4042513

[300] H. Schempf, "Neptune: above-ground storage tank inspection robot system," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 1403–1408 vol.2.

[301] P. McGarey, M. Polzin, and T. D. Barfoot, "Falling in line: Visual route following on extreme terrain for a tethered mobile robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2027–2034.

[302] M. Krishna, J. Bares, and E. Mutschler, "Tethering system design for dante ii," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, 1997, pp. 1100–1105 vol.2.

[303] K. Nagatani, S. Tatano, K. Ikeda, A. Watanabe, and M. Kuri, "Design and development of a tethered mobile robot to traverse on steep slope based on an analysis of its slippage and turnover," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2637–2642.

[304] M. A. Sebok and H. G. Tanner, "On the hybrid kinematics of tethered mobile robots," in *2019 American Control Conference (ACC)*, 2019, pp. 25–30.

[305] T. Rasheed, P. Long, and S. Caro, "Wrench-Feasible Workspace of Mobile Cable-Driven Parallel Robots," *Journal of Mechanisms and Robotics*, vol. 12, no. 3, 01 2020, 031009. [Online]. Available: https://doi.org/10.1115/1.4045423

[306] S. Lupashin and R. D'Andrea, "Stabilization of a flying vehicle on a taut tether using inertial sensing," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2432–2438.

[307] E. F. Fukushima, N. Kitamura, and S. Hirose, "Development of tethered autonomous mobile robot systems for field works," *Advanced Robotics*, vol. 15, no. 4, pp. 481–496, 2001. [Online]. Available: https://doi.org/10.1163/156855301750398374

[308] S. Kim, S. Bhattacharya, and V. Kumar, "Path planning for a tethered mobile robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1132–1139.

[309] S. Kim and M. Likhachev, "Path planning for a tethered robot using multi-heuristic a* with topology-based heuristics," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4656–4663.

[310] J. Iqbal, S. Heikkila, and A. Halme, "Tether tracking and control of rosa robotic rover," in *2008 10th International Conference on Control, Automation, Robotics and Vision*, 2008, pp. 689–693.

[311] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 722–728.

[312] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 403–410, 1988.

[313] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 410–425, 1991. [Online]. Available: https://doi.org/10.1177/027836499101000409

[314] S. Chiaverini, G. Oriolo, and I. D. Walker, *Kinematically Redundant Manipulators*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 245–268. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_12

[315] H. Simas and R. Di Gregorio, "A Technique Based on Adaptive Extended Jacobians for Improving the Robustness of the Inverse Numerical Kinematics of Redundant Robots," *Journal of Mechanisms and Robotics*, vol. 11, no. 2, 02 2019, 020913. [Online]. Available: https://doi.org/10.1115/1.4042514

[316] Y. Nakamura and H. Hanufasa, "Task priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[317] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[318] B. Siciliano and J. . E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 1991, pp. 1211–1216 vol.2.

[319] S. Chiaverini, "Task-priority redundancy resolution with robustness to algorithmic singularities," *IFAC Proceedings Volumes*, vol. 27, no. 14, pp. 453 – 459, 1994, fourth IFAC Symposium on Robot Control, Capri, Italy, September 19-21, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017473526

[320] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[321] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 1, 1998, pp. 323–329 vol.1.

[322] A. Leigeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 12, pp. 868–871, December 1977.

[323] R. V. Dubey, J. A. Euler, and S. M. Babcock, "An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 28–36 vol.1.

[324] B. E. Bishop and D. J. Stilwell, "On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204)*, 2001, pp. 823–828.

[325] B. E. Bishop, "On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 5, pp. 608–615, 2003.

[326] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester, "Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 4, pp. 842–848, 2005.

[327] G. Antonelli and S. Chiaverini, "Kinematic control of a platoon of autonomous vehicles," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 1, 2003, pp. 1464–1469 vol.1.

[328] G. Antonelli and S. Chiaverini, "Obstacle avoidance for a platoon of autonomous underwater vehicles," *IFAC Proceedings Volumes*, vol. 36, no. 21, pp. 115 – 120, 2003, 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2003), Girona, Spain, 17-19 September, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017377935

[329] M. Caruso, P. Gallina, and S. Seriani, "On the modelling of tethered mobile robots as redundant manipulators," *Robotics*, vol. 10, no. 2, 2021.

[330] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, UK: Springer-Verlag, 2009.

[331] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. [Online]. Available: https://doi.org/10.1177/0278364911406761

[332] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 732–738.

[333] S. Seriani, P. Gallina, and A. Wedler, "A modular cable robot for inspection and light manipulation on celestial bodies," *Acta Astronautica*, vol. 123, pp. 145 – 153, 2016, special Section: Selected Papers from the International Workshop on Satellite Constellations and Formation Flying 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576516300388

[334] R. Hercik, R. Byrtus, R. Jaros, and J. Koziorek, "Implementation of autonomous mobile robot in smartfactory," *Applied Sciences*, vol. 12, no. 17, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/17/8912

[335] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng'oma, W. Liu, P. G. Mcbride, M. G. Shultz, C. Cerasi, and D. Sun, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5g-based smart manufacturing applications," *IEEE Access*, vol. 8, pp. 202 312–202 353, 2020.

[336] G. Lemasurier, G. Bejerano, V. Albanese, J. Parrillo, H. A. Yanco, N. Amerson, R. Hetrick, and E. Phillips, "Methods for expressing robot intent for human–robot collaboration in shared workspaces," *J. Hum.-Robot Interact.*, vol. 10, no. 4, sep 2021. [Online]. Available: https://doi.org/10.1145/3472223

[337] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 797–803.

[338] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2153–2160.

[339] A. Abdulov and A. Abramenkov, "Collision avoidance by communication for autonomous mobile robots in crowd," in *2018 Eleventh International Conference "Management of large-scale system development" (MLSD*, 2018, pp. 1–4.

[340] S. Seriani, L. Marcini, M. Caruso, P. Gallina, and E. Medvet, "Crowded environment navigation with neat: Impact of perception resolution on controller optimization," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 101, no. 2, 2021.

[341] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[342] K. Dimitropoulos, I. Hatzilygeroudis, and K. Chatzilygeroudis, "A brief survey of sim2real methods for robot learning," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2022, pp. 133–140.

[343] S. Yang, T. Li, X. Gong, B. Peng, and J. Hu, "A review on crowd simulation and modeling," *Graphical Models*, vol. 111, p. 101081, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1524070320300242

[344] ——, "A review on crowd simulation and modeling," *Graphical Models*, vol. 111, p. 101081, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1524070320300242

[345] T. Fraichard and V. Levesy, "From crowd simulation to robot navigation in crowds," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 729–735, 2020.

[346] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep 2000. [Online]. Available: https://doi.org/10.1038/35035023

[347] D. Helbing and A. Johansson, *Pedestrian, Crowd and Evacuation Dynamics*. New York, NY: Springer New York, 2009, pp. 6476–6495. [Online]. Available: https://doi.org/10.1007/978-0-387-30440-3_382

[348] I. Karamouzas, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Phys. Rev. Lett.*, vol. 113, p. 238701, Dec 2014. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.113.238701

[349] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[350] A. Mirhoseini, A. Goldie, M. Yazgan, J. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, S. Bae *et al.*, "Chip placement with deep reinforcement learning," *arXiv preprint arXiv:2004.10746*, 2020.

[351] M. G. Bellemare, S. Candido, P. S. Castro, J. Gong, M. C. Machado, S. Moitra, S. S. Ponda, and Z. Wang, "Autonomous navigation of stratospheric balloons using reinforcement learning," *Nature*, vol. 588, no. 7836, pp. 77–82, 2020.

[352] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[353] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[354] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[355] K. Katyal, Y. Gao, J. Markowitz, I.-J. Wang, and C.-M. Huang, "Group-aware robot navigation in crowded environments," 2020.

[356] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[357] P. Regier, I. Shareef, and M. Bennewitz, "Improving navigation with the social force model by learning a neural network controller in pedestrian crowds," in *2019 European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–6.

[358] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[359] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[360] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[361] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLOS ONE*, vol. 5, no. 4, pp. 1–7, 04 2010. [Online]. Available: https://doi.org/10.1371/journal.pone.0010047

[362] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.51.4282

[363] A. Johansson, D. Helbing, and P. Shukla, "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data," 2008.

[364] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings. Biological sciences / The Royal Society*, vol. 276, pp. 2755–62, 06 2009.

[365] M. Caruso, E. Regolin, F. J. Camerota Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, "Robot navigation in crowded environments: A reinforcement learning approach," *Machines*, vol. 11, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2075-1702/11/2/268

[366] M. Caruso, E. Regolin, F. J. C. Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, "Robot navigation in crowded environments: a reinforcement learning approach," *Preprints*, Dec. 2022. [Online]. Available: https://doi.org/10.20944/preprints202212.0233.v1

[367] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[368] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan *et al.*, "Ray: A distributed framework for emerging {AI} applications," in *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 2018, pp. 561–577.

[369] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," *Machine learning*, vol. 16, no. 3, pp. 185–202, 1994.

[370] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," in *Annual conference on artificial intelligence*. Springer, 2011, pp. 335–346.

[371] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[372] L. Žlajpah and T. Petrič, "Obstacle avoidance for redundant manipulators as control problem," in *Serial and Parallel Robot Manipulators*, S. Kucuk, Ed. Rijeka: IntechOpen, 2012, ch. 11. [Online]. Available: https://doi.org/10.5772/32651

[373] H. Shen, H. Wu, B. Chen, Y. Jiang, and C. Yan, "Obstacle avoidance algorithm for 7-dof redundant anthropomorphic arm," *Journal of Control Science and Engineering*, vol. 2015, pp. 1–9, 01 2015.