Politecnico di Torino
ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Mechanical Engineering (36thcycle)

Robotics for Electric Waste Recycling: Vision-based algorithms for safe Human-Robot collaboration

By

Laura Salamina

******

Supervisor(s):
Prof. Stefano Mauro

Doctoral Examination Committee:
Prof. Pierluigi Beomonte Zobel, Referee, Università degli Studi de L'Aquila
Prof. Monica Malvezzi, Referee, Università degli Studi di Siena

Politecnico di Torino
2024

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Laura Salamina
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

To my beloved family

Abstract

In recent years, sustainable waste management has become a critical focus, emphasizing the need for advanced technological solutions within the circular economy framework. The transition towards a circular economy aims to significantly reduce waste by continually using resources through design principles that emphasize longevity, renewability, reuse, repair, re-manufacturing, and recycling. This paradigm shift is crucial for addressing resource scarcity and environmental sustainability. Robotics is contributing to this transition, enabling precise and efficient waste sorting and processing.

This work explores the integration of vision-based algorithms in a collaborative robotic workcell, specifically targeting the disassembly of large electric and electronic waste. The objective is to enhance the disassembly process by employing robots working together with human operators, improving disassembly efficiency. A case study of an electric vehicle battery was examined to identify the disassembly tasks and determine the optimal robotic solution for these activities. This analysis led to the selection of a mobile manipulator as the most suitable solution for the disassembly process, and to the identification of vision systems for the control algorithms of the robots. A dynamic model of the robotic system was developed to monitor the interaction between the two robots and their movements within the workspace and the collaborative cell. Particular attention was given to developing the model for the mobile robot, and to the forces exchanged between the wheels and the floor. The research delved into the development and implementation of collaborative robotics algorithms, including gaze tracking for robot control and collision avoidance algorithms. A gaze tracking system was studied and tested to enhance human-robot interaction by enabling the robot to respond to the operator visual focus. The collision avoidance algorithms were developed for both the mobile robot and the anthropomorphic robotic arm, ensuring safe and efficient operation within the collaborative workspace. These algorithms were tested in simulation environments and validated through experiments using a TurtleBot for the mobile robot, and a UR5 robotic arm.

The optimal mobile robot and robotic arm were chosen, together with the vision systems. The practical implementation of the collaborative robotic cell for electric vehicle battery disassembly were carried out through detailed experimental setups and tests. This included the integration of vision systems, precise localization techniques, and effective collision avoidance algorithms. The experiments were divided into three main categories: collision avoidance between the mobile robot and the human operator, precise positioning of the robotic arm relative to the workbench, and collision avoidance between the robotic arm and the human operator. The results from these tests demonstrated the system potential to support complex disassembly tasks safely in industrial settings. In the final experimental phase, the focus was on the practical application of the developed algorithms in a real-world scenario. The mobile robot's path was optimized to avoid collisions with human operators, in a predictable way, and the anthropomorphic robotic arm was tested for its ability to continue tasks without interruption despite human interference. The effectiveness of the localization algorithms was verified by the robot capability to accurately position itself and perform precise tasks.

The results confirm that the combination of advanced vision-based control and collaborative robotics can significantly enhance the efficiency and safety of large electric waste disassembly processes. They also demonstrate the applicability of these technologies for performing operations that cannot be fully automated on large-scale systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The circular economy represents a systemic shift from the linear ’take-make-waste’ model to an economic system aimed at eliminating waste and the continual use of resources. Central to this economy is the decoupling of economic activity from the consumption of finite resources and designing waste out of the system. It is predicated on the principles of designing for longevity, renewability, reuse, repair, remanufacturing, and recycling - ensuring that products and materials continually circulate in the economy [1]. In figure 1.1, a schematics of circular economy operation is shown. In addressing resource scarcity, the circular economy is not only an economic model but a response to the environmental imperative. By keeping products and materials in use, it strives to preserve natural capital, optimize resource yields, and minimize system risks by managing finite stocks and renewable flows. This is where waste management becomes crucial: it turns from a service dealing with waste to a value-creating activity in the circular system. Through advanced sorting, collection, and recycling technologies, waste management is transformed into resource management, closing the loop of material flows [2]. Technological innovation, particularly in the field of robotics and AI, has accelerated the transition to a circular economy by enabling the sorting and processing of waste streams with unprecedented precision and efficiency. In waste management, robotics provide solutions for automated sorting which reduces contamination and increases the value of recovered materials. AI-driven analytics can optimize the collection and processing pathways, minimizing the environmental impact and cost of recycling [3]. The shift to a circular economy thus presents a dual advantage: it offers a strategy for long-term sustainability while also providing economic opportunities through

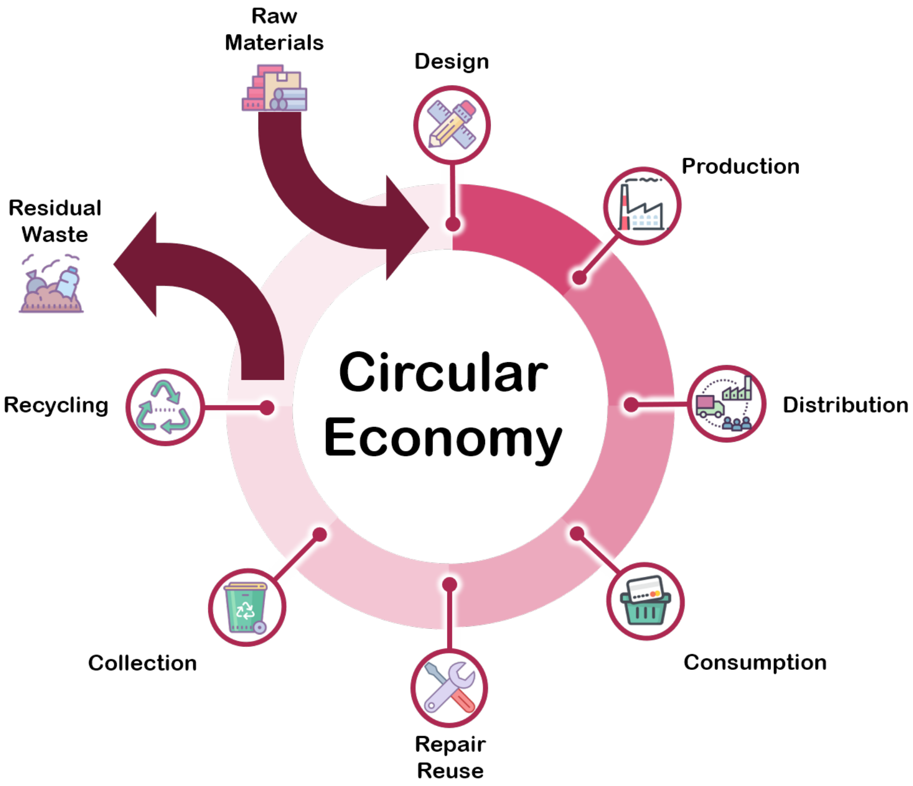


Fig. 1.1 Circular economy cycle

the recovery and regeneration of products and materials. The potential for growth in this sector is significant, with innovation in waste management being a key driver for the circular economy. Within the framework of the circular economy, the recycling of electrical and electronic materials holds a pivotal role, given that these materials contain valuable metals and significant components. Their effective reintegration into the production cycle through advanced recycling processes can substantially reduce the reliance on virgin resources and decrease the environmental impact associated with the extraction and processing of new raw materials. The recycling of these materials not only supports environmental sustainability objectives but also aligns perfectly with the principles of the circular economy, which aims to design systems where waste is reduced through the reintegration of materials into the economy. Through innovative sorting and processing technologies, it is possible to maximize the value and utilization of recovered materials, closing the loop of material flows and contributing to the creation of a more sustainable economic system.

1.1 Aim of the work

The present dissertation introduces some methodologies to enhance the semi-automated disassembly of electric and electronic waste. Within this context, it is most relevant the use of vision-based algorithms for robot control and their integration in a collaborative robotic workcell. The development and implementation of these algorithms showcases the potential for a collaborative robotic cell that is capable of performing assembly and disassembly tasks on large and complex mechanical systems. The focus on electric waste disassembly addresses a pressing need for sustainable recycling practices. A comprehensive study of the state of the art revealed that the application of collaborative robotics is particularly well-suited to the disassembly of electric vehicle batteries. This is due to the complexity and the need for flexibility in these tasks, which are difficult to fully automate. Consequently, electric vehicle batteries were chosen as a representative example of electric and electronic waste for this research. The rapid growth in electric vehicle production underscores the importance of efficient battery recycling methods to mitigate environmental impact and recover valuable materials. The specific objectives of this thesis are developed after an initial comprehensive review of advanced robotic solutions for waste management. First, an in-depth analysis of the disassembly process for electric vehicle batteries is conducted, identifying key tasks and robotic solutions for these activities, which is presented in chapter 2. Dynamic models of robotic systems, including mobile robots and robotic arms, are then developed and validated to ensure accurate simulations, with a comprehensive state of the art introduced in chapter 3. Next, vision-based algorithms are presented, with a comprehensive state of the art, to allow human robot collaboration in the disassembling process, detailed in chapter 4. Finally, collision avoidance algorithms for both mobile robots and robotic arms are tested both in a simulation environment and in laboratory to ensure safe operation within a collaborative workspace, explored in chapter 5.

This thesis contributes to the field by presenting algorithms that are not only innovative but also practical for real-world applications. A dynamic model of the robotic system, as well as a simulation environment comprising the robots and a human operator, was developed as a tool to support the experimental activity. The proposed collaborative robotic cell demonstrates the feasibility of using robotic systems for the disassembly and assembly of large mechanical systems, a critical step towards the automation of recycling processes for electric and electronic waste. Through this work, the aim is to push the boundaries of what is possible in collaborative robotics, setting the stage for future research and application in industrial automation.

1.2 State of the Art: Robotics for waste management

In this section, an overview of advanced robotic solutions for waste management and recycling is presented, highlighting how the adoption of digital and intelligent technologies can revolutionize these practices. This overview provides a broader context of the research field and sets the stage for the specific focus of the thesis. The main objectives of this review are:

- Optimization of Waste Management through Digital Technologies: the use of IoT devices, artificial intelligence, and robotics to improve sorting, processing, and re-manufacturing activities.
- Enhancement of Recycling Efficiency: the impact of automated waste sorting systems, which utilize sensor arrays and conveyor belts to separate waste materials, reducing human intervention and increasing sorting speed.
- Introduction of Autonomous Robotic Solutions: employing autonomous robots for collecting and sorting waste directly from the source, optimizing collection routes and ensuring comprehensive urban area coverage.
- Integration of AI and ML in Waste Management: how the use of artificial intelligence and machine learning in waste management robots can improve the identification and classification of various waste types, handling uncertainties in waste material characteristics.
- Application of Smart Technologies in the Circular Economy: how digitization and intelligent robotics can create a closed-loop system, where waste materials are efficiently recycled and reintroduced into the economy.
- Collaborative Robotics for Complex Disassembly Tasks: the role of collaborative robotics in the disassembly of complex electronic and electric waste, focusing on human-robot interaction to enhance efficiency and safety.

This section provides a general overview of the state of the art in robotics for waste management. The detailed exploration of collaborative robotics and the dynamic modeling of robotic systems will be presented in subsequent chapters, specifically focusing on their application in the disassembly of electric vehicle batteries.

In recent years, the imperative for sustainable waste management practices has become increasingly evident, driving the exploration of innovative solutions at the intersection of technology and environmental stewardship. The advent of Industry 4.0, characterized by the integration of digital technologies and intelligent robotics into manufacturing and production processes, presents unprecedented opportunities to advance the principles of the circular economy [4]. This economy emphasizes the reduction, reuse, and recycling of materials to minimize waste and promote resource efficiency.

[5] and [6] collectively offer a comprehensive exploration of how digital transformation and smart technologies are reshaping waste management and recycling practices, underlining a pivotal shift towards Recycling 4.0, where automation, data analytics, and connectivity play crucial roles in enhancing the efficiency and effectiveness of recycling processes.

The integration of digital technologies [7] could facilitate the creation of intelligent waste management systems capable of optimizing resource use, streamlining recycling operations, and reducing environmental impact. These systems can leverage Internet of Things (IoT) devices, artificial intelligence, and robotics to improve sorting, processing, and re-manufacturing activities, ensuring that materials are efficiently returned to the production cycle. Furthermore, the application of big data and analytics enables better decision-making, process optimization, and innovation in product design and material usage, aligning with the principles of a circular economy.

However, the transition to Recycling 4.0 and the broader adoption of circular economy practices also present challenges, including the need for significant investments in technology, the development of new competencies, and the establishment of supportive regulatory frameworks. The success of this transition relies on multi-stakeholder collaboration, including policymakers, industry leaders, and the scientific community, to foster an ecosystem that supports sustainable practices and leverages technological advancements for environmental conservation.

Recycling municipal solid waste (MSW) has evolved significantly with the advent of robotic and automated solutions, marking a transformative approach to managing urban waste sustainably. These innovative technologies aim to enhance the efficiency and effectiveness of recycling processes, addressing the growing environmental concerns associated with waste management. For instance, automated waste sorting systems, as described in [8] employ sensor arrays and conveyor belts to segregate waste materials like steel cans, aluminum cans, glass bottles, and plastics with high accuracy. This automation minimizes human intervention and increases the sorting speed, contributing to a more efficient recycling process.

Robotic solutions, such as those outlined in [9] introduce autonomous robots capable of collecting and sorting waste directly from the source. These robots operate in swarms, optimizing waste collection routes and ensuring comprehensive urban area coverage. This approach not only reduces the labor intensity of waste collection but also enhances the segregation accuracy, crucial for effective recycling.

Furthermore, the integration of Artificial Intelligence (AI) and Machine Learning (ML) in waste management robots, as explored in [10] enables these machines to identify and classify various waste types. The use of fuzzy logic control systems allows for the handling of uncertainties in waste material characteristics, leading to improved sorting precision and adaptability in dynamic waste compositions.

Another innovative approach is the application of digitization and intelligent robotics in the circular economy [6]. This concept leverages smart technologies to create a closed-loop system, where waste materials are efficiently recycled and reintroduced into the economy, minimizing waste generation and resource consumption.

Each of these solutions brings unique advantages to municipal solid waste recycling. Automated sorting systems increase throughput and reduce contamination in recyclables, while robotic collection enhances operational efficiency and coverage. AI and ML integration into these systems further refine sorting accuracy, adapting to the variability of waste streams. Collectively, these robotic and automated solutions represent a significant leap towards achieving sustainable and efficient waste management and recycling processes, aligning with the goals of the circular economy and environmental conservation.

1.2.1 Robotics in the Recycling of WEEE

The management and recycling of Waste Electrical and Electronic Equipment (WEEE) present critical environmental and economic challenges globally. WEEE encompasses a broad spectrum of discarded electronic devices ranging from household appliances to IT and telecommunications equipment. The accelerated pace of technological advancements and consumer demand for the latest gadgets have significantly contributed to the exponential increase in electronic waste (e-waste).

In 2016, the European Union witnessed an average e-waste generation of 16 kg per person [11]. This alarming increase underscores the importance of developing circular economy models that prioritize recycling and reuse over landfilling, aiming to reduce the environmental footprint and recover valuable materials. The European directives on WEEE [12] and the restriction of hazardous substances (RoHS) [13] are pivotal legislative measures that seek to enhance waste management and recycling capacities, thereby addressing the challenges posed by e-waste. This surge necessitates effective recycling strategies to mitigate environmental hazards and reclaim valuable materials [14]. However, the complex composition of WEEE, along with the rapid technological advancements, poses significant challenges to traditional recycling methods [11, 15]. This has necessitated the exploration of innovative solutions, among which robotic technologies have emerged as a game-changer in the domain of WEEE recycling.

The integration of robotics into disassembly processes emerges as a critical component for sustainable manufacturing and waste management within the principles of Industry 4.0. Poschmann [16] provides a comprehensive overview, underscoring the necessity for both applied and fundamental research in this area. This review highlights the exploration of predefined processes alongside adaptable, flexible automation techniques. Significant advancements in robotic technologies, including Artificial Intelligence (AI) and the Internet of Things (IoT), present promising prospects for automating disassembly tasks. Notably, Apple Inc.'s development of robotic recycling and disassembly machines, Daisy and Liam, exemplify practical demonstrations of robotic disassembly cells, marking a transition towards more efficient, sustainable, and automated processes in re-manufacturing and recycling industries.

Furthering the discourse on efficiency enhancement in recycling end-of-life electronic products, some innovative approaches for investigating the disassembly cycle are explored. In [17], a model that combines decision-makers' preferences with advanced algorithms to optimize the disassembly process is presented. This solution enables to minimize uncertainties and maximize material recovery. Employing a combination of Analytic Hierarchy Process (AHP) and Simulated Annealing (SA), the study showcases a system capable of adapting to changes in product composition, thus improving recycling operations' effectiveness. The collective works of Li [18–20] explore automated and robotic disassembly approaches for recycling components from end-of-life products, particularly focusing on electric vehicle (EV) components. These studies highlight the shift towards more sustainable recycling methods by leveraging automation to enhance the efficiency and economic viability of the recycling process. Many studies have posed their attention on the disassembly of small electronic components. Marconi et al. [21] delve into the development of a robotic system for the automated disassembly of electronic components from end-of-life electric boards. Their work assesses the economic and environmental feasibility of reusing electronic components, aiming to enhance sustainability in the electronics sector. The prototyped disassembly system, adapted from a wave soldering machine and complemented by a two-axis manipulator with a suction cup, proved effective in experimental tests, with all disassembled microprocessors being reusable without further treatment. In the field of e-waste management, Nowakowski et al. [22] introduces a deep learning-based system to improve e-waste collection planning through image recognition. By enabling users to photograph and upload images of e-waste, the system employs convolutional neural networks (CNNs) and region-based CNNs (R-CNNs) for accurate classification and size determination of waste items. This approach enhances the efficiency of e-waste collection planning, allowing for precise vehicle allocation and capacity planning. The system demonstrates high accuracy rates between 90% to 97% in recognizing and classifying various e-waste categories, facilitating a more efficient and effective e-waste management process. In the same way, Babbitt et al. [23] focus on the formulation of an exhaustive database from the meticulous disassembly of 95 consumer electronics. This database aims to illuminate sustainable lifecycle management practices, bridging the gap between theoretical Bills of Materials (BOMs) and the pragmatic conditions observed at the end-of-life phase of electronic products, due to deteriorative states products invariably reach at their end of life. Such endeavors advocate for a re-calibration of recycling strategies towards principles of circular economy and sustainability, and underlines how important is to be able to recognize a successful strategy for disassembly, especially when the conditions of the components do not match with the datasheet ones. It turns out that, human-robot collaboration (HRC) can be a good solution for the disassembly of WEEE components. Addressing micro-robotic handling solutions for printed circuit board (PCB) re-manufacturing, Ruggeri et al. [24] emphasize the synergy between human expertise and robotic precision. The integration of tools like vacuum micro-grippers and solder micro-ball sorting devices into flexible work-cells not only enhances efficiency but also fosters collaborative work environments where humans and robots complement each other's capabilities. The possibility to integrate robots in the disassembly process allows to address repetitive and physically harmful tasks to robots, while decisional and more sophisticated activities to the operator. Hence, Chatzikonstantinou et al. [25] and Li [26] elaborate on methods to enhance collaboration between humans and robots in the disassembly process. These approaches aim to optimize the allocation of disassembly tasks, focusing on efficiency, adaptability, and the incorporation of human factors such as fatigue, thereby ensuring more ergonomic manufacturing processes.

Alvarez [14] and Renteria [27] discuss the integration of HRC in recycling electronic waste. Emphasizing the optimization of recycling processes through technical and economic criteria, these works highlight the benefits of HRC in managing electronic waste's complexity and variability. This collaborative approach not only renders recycling processes more efficient but also improves job satisfaction by allocating hazardous tasks to robots and complex tasks to humans.

In conclusion, the necessity of utilizing robotics in collaboration with humans for the disassembly of WEEE, stands out as imperative for advancing towards sustainable recycling practices. This approach balances technological innovation with human expertise to improve both environmental and economic outcomes, emphasizing a transition towards more efficient, sustainable, and automated processes in the recycling industry.

1.2.2 Disassembly and Recycling of Electric Vehicle Batteries

Electric vehicle battery (EVB) recycling represents a pivotal challenge for the automotive industry. With the rapid expansion of the electric vehicle market, fueled by increasing awareness of environmental issues and the urgency to reduce greenhouse gas emissions, the need to develop sustainable solutions for the life-cycle management of lithium batteries becomes paramount. Despite significant advancements in the automation of battery production in recent years, the approach to their end-of-life (EoL) treatment remains relatively simplistic, based on manual disassembling.

The recycling of the components of electric and hybrid vehicles needs for efficient and environmentally sustainable methods to manage new and critical materials like rare earth elements, cobalt, antimony, and palladium. However, there is a lack of established industrial-scale recycling pathways for traction batteries, electric motors, and power electronics, exacerbated by the diversity in vehicle designs and component configurations [28]. The high concentrations of strategic metals as lithium and cobalt in EVB packs make the latters an attractive sources for metal recovery, as they often boast significantly higher metal concentrations than those found in natural ores, enhancing their appeal for recycling efforts [29, 30]. Besides, it is fundamental to define cost-effective and energy-efficient recycling solutions for EoL EVB, to mitigate environmental impacts, reduce reliance on raw material extraction, and address safety and contamination risks [31].

[32] evaluates the environmental impact and supply chain logistics of recycling electric vehicle (EV) batteries within California's context. Highlighting the urgent need for an advanced recycling infrastructure to adeptly handle the lifecycle end of lithium-ion batteries, this research integrates life-cycle assessment with geographic information systems. This integration facilitates a nuanced analysis of energy consumption, greenhouse gas emissions, water use, and air pollution across various recycling methods, such as hydrometallurgical and pyrometallurgical processes. Furthermore, it explores the socio-economic and environmental advantages of strategically situating battery dismantling and recycling centers, utilizing geospatial modeling to identify the most beneficial locations. A regulatory proposal from the European Commission [33] aims at promoting a battery recycling industry in Europe, with a particular attention to EVB. [34] analyzes the role of electric mobility in reducing CO_2 emissions, the impact of lithium-ion battery production on the environment and industry, and the European Commission's proposal for recycling thresholds in batteries made after 2030. It uses a material flow model to assess the viability of these recycling targets, taking into account battery life, technological advancements, and recycling efficiency. In addressing the challenges of sustainable battery usage in electric vehicles, [35] presents a compelling argument for the re-manufacturing of lithium-ion battery modules. This research proposes a systematic methodology for redesigning battery modules to facilitate automated re-manufacturing processes, thereby maintaining the module integrity and extending its life-cycle. This approach highlights the importance of considering both product design and manufacturing processes in achieving sustainable re-manufacturing solutions.

The adoption of robotic technologies, such as artificial vision systems, robotic manipulators, and machine learning platforms, promises to revolutionize the battery disassembly process, making it more efficient, safe, and less costly compared to traditional methods. A detailed techno-economic analysis of robotic disassembly technologies versus other recycling approaches highlights significant advantages in terms of operational cost reduction and environmental impact. The recycling of EV batteries must evolve in tandem with battery technology advancements to address the fluctuating costs of materials, distribution and production challenges, and the environmental implications of transport and recycling processes. The pursuit of novel recycling methods, including enhancements in robotic disassembly, hydrometallurgical, and pyrometallurgical processes, is essential for ensuring the sustainable supply of critical materials and mitigating the environmental impact of spent lithium-ion batteries (LIBs) [29, 31].

The industrial disassembly of EV batteries faces technical, economic, and environmental challenges. A proficient dismantling process is pivotal for the extraction of valuable materials, diminishing the environmental footprint, and supporting sustainable advancements within the automotive sector.

The rapid increase of EoL EV batteries asks for finding solutions for their usage after they are collected from vehicles. A possible option is re-manufacturing, as proposed by [36]. Re-manufacturing can be seen as a good option for a battery after first life cycle, as battery reaches its EoL when its capacity is the 80% of the original one. Re-manufacturing involves restoring EoL EV batteries to like-new condition with a warranty for the buyer, with complete disassembly, thorough cleaning, examination for damage, and reprocessing to original equipment manufacturer (OEM) specifications. The prediction of reaching EoL is not trivial, as the battery performance depends on the season weather where the EV moves, and on the use the EV owners make of the car. This determines an uncertainty of the state of health (SOH) of the battery when the EV reaches the EOL. The SOH is the residual energy capacity with respect to the original one. According to [37], the EV battery SOH at their EOL can be assumed as normal distribution, as the one shown in figure 1.2. This implies that batteries reach the collection facility with different values of SOH. When batteries reach collecting facilities, they are checked for ensuring they are not damaged. If so, batteries should be directly recycled. For the other batteries it is necessary to check their SOH to establish their possible second life applications [37]:

- if SOH is above 88%, the battery can be send back to a first life as a spare part to replace damaged or older batteries;
- if SOH is between the 75% and the 88%, the battery can be used, as it is, for stationary applications;

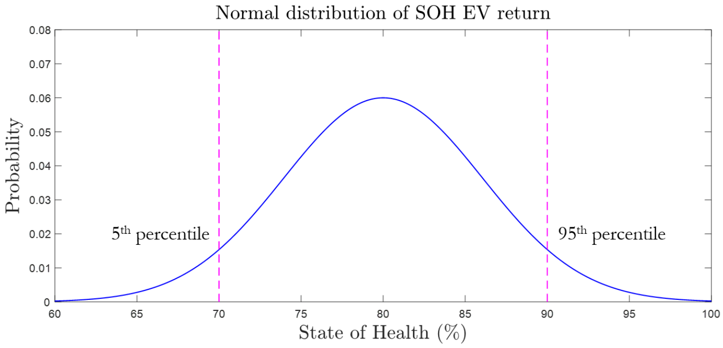


Fig. 1.2 Assumed normal distribution for state of health of electric vehicle return, from [37].

- if SOH is lower than 75%, the battery can be dismantled in modules and/or cells and used for smaller devices.

In [38], the potential of repurposed EV batteries for grid services like energy arbitrage and demand peak shaving is highlighted. The work reveals how the performance of EV batteries in second-life applications varies among manufacturers, influenced by original design factors such as thermal management and energy density. The research suggests repurposing whole EV battery packs, when their health allows, as an effective, sustainable option for new utility-scale energy storage, providing criteria for battery selection based on performance metrics for second-life uses. The modules and the cells inside an EVB do not age uniformly in a battery, because of the uneven temperature distribution inside the battery itself. The SOH of a battery depends mostly on the weakest cells in a battery. Therefore, even when the SOH of a battery results very low, it does not mean that the complete battery system is useless and should be scrapped. So, to enhance the lifespan of an EV battery, either entire modules or individual cells can be replaced, maintaining its original application's efficiency. Alternatively, well-functioning parts from an EoL battery pack can be repurposed for uses with lower performance demands, like stationary energy storage. These approaches necessitate methodical disassembly. Current recycling practices involve basic manual dismantling up to the module level, followed by pyrometallurgical or hydrometallurgical processes. A more detailed disassembly to the cell level is essential for direct recycling of active materials, optimizing resource recovery. However, second-life strategies for batteries enhance their lifespan without competing with EoL recycling; they extend the operational life of battery systems, modules, or cells. After their extended use, all batteries will be recycled, ensuring a sustainable closed-loop system [39]. The criticality of advanced disassembly for facilitating efficient recycling and repurposing of EV batteries is underscored in the recent studies in [39–42], which explore the complexities and advancements in both pyrometallurgical and hydrometallurgical recycling methods, as well as the promising avenue of direct recycling for specific battery chemistries. Pyrometallurgical processes are utilized for treating various battery chemistries simultaneously, mainly targeting the extraction of valuable metals such as cobalt, nickel, and copper, while prioritizing high processing volume. However, due to economic factors, elements like aluminum and lithium often remain unrecycled within the slag. Following pyro-metallurgy, the resultant mix of metals can be either downcycled as alloys or further processed through hydrometallurgy to separate and recover these valuable components. Pyrometallurgical processes are actually the predominant recycling processes, as EoL battery stream involves also cells from small electronic devices and pyrometallurgy is able to handle different cell types. Hydrometallurgical processes demand more meticulous disassembly and specific sorting by battery chemistry due to their sensitivity to impurities. These methods also require the extensive use of various reagents, making them resource-heavy. Despite this, they enable the recovery of not only cathode materials but also high-grade lithium carbonate and a greater quantity of aluminum. Another possible recycling process is direct recycling. Direct recycling has to be adapted to each battery chemistry. The process focuses on the extraction of active materials that can be directly reintroduced into cell production. While the direct recovery of active materials saves energy and resources in production and leads to higher efficiency levels, at the same time it has high requirements regarding homogeneity of the cell chemistry of the processed batteries. Due to the fact that predominant active materials as well as their respective material ratio change over time as a result of technological development of cell chemistry, direct recycling is particularly attractive for production waste and lower aged batteries as the recovered active materials can directly be reintroduced into the production process [39].

Figure 1.3 illustrates the comprehensive supply chain for battery production and how disassembly facilitates additional circular economy pathways for electric vehicle batteries. In the dashed boxes delineate various sectors and companies that participate in the circular supply chain, highlighting their roles in a comprehensive closed-loop system. As can be seen, the supply chain starts from raw material extraction, following with the production of the batteries, that are integrated in the EVs. Then, the battery enters the use phase in its first life.

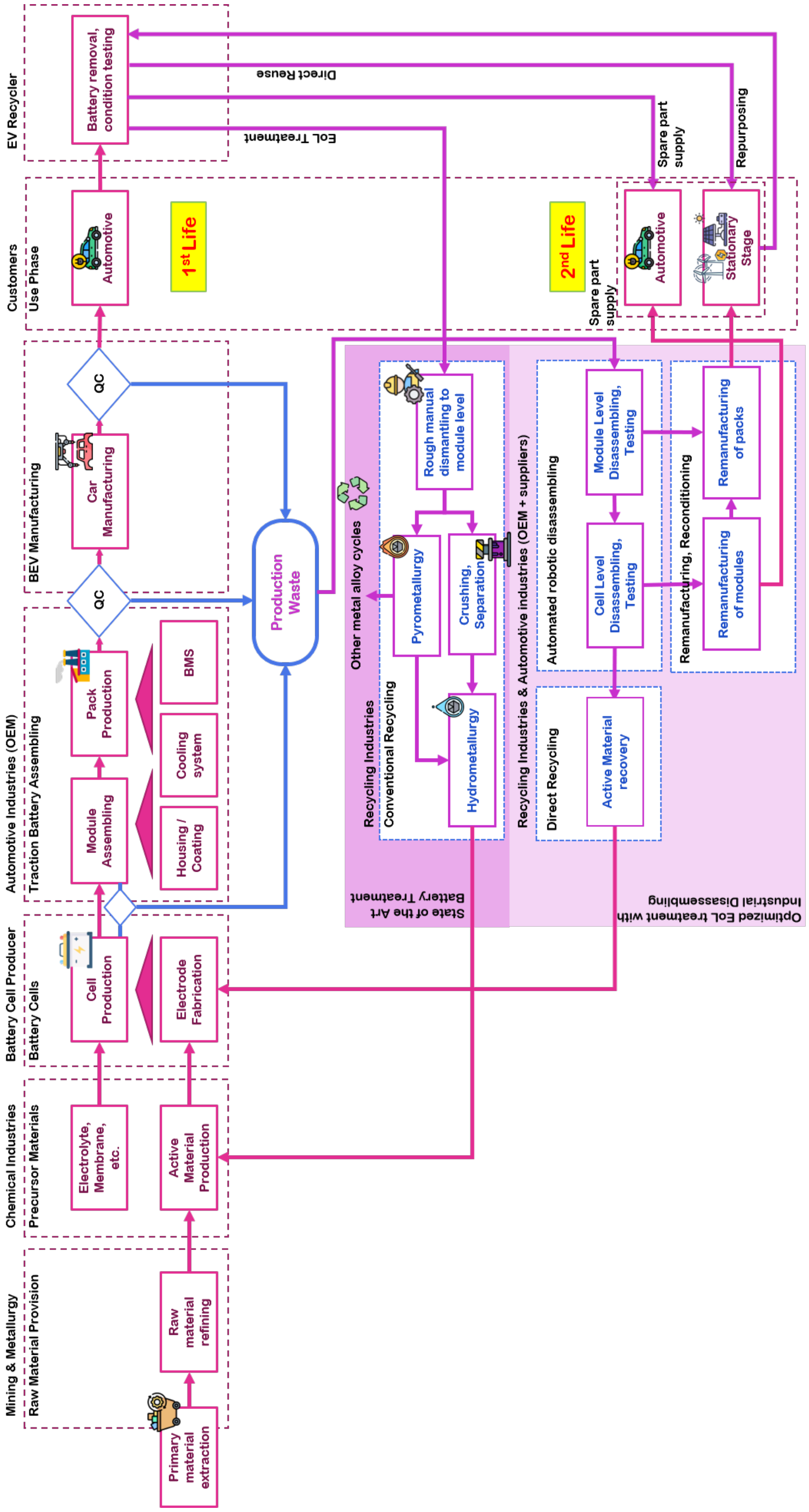


Fig. 1.3 EV battery production supply chain and disassembling role in fostering circular economy pathways, from [39]

Later on, the battery is collected and a decision making process is necessary to address EoL battery in second life phase or in a recycling process. Disassembling EV batteries is crucial for EoL strategies, enabling repurposing, re-manufacturing, and facilitating efficient recycling. Manual disassembly, however, incurs high costs and safety risks, including thermal runaway and explosion hazards. Advanced disassembly techniques allow for precise separation to module or cell levels. With the rise in EV sales, manual methods may hinder the EoL battery treatment process in the future, underscoring the need for improved disassembly strategies. [43] examines the diversity of electric vehicle (EV) batteries, highlighting the factors that contribute to differences among them.

It categorizes EV batteries based on attributes such as the chemistry of cells, cell types, configurations, and the types of cooling systems employed. In particular:

- Cell chemistry: within EV batteries, diverse cell chemistries such as Lithium Manganese Oxide (LMO), Lithium Iron Phosphate (LFP), and Nickel Manganese Cobalt (NMC), Lithium Nickel Cobalt Aluminium Oxide (NCA), Lithium Cobalt Oxide (LCO), Lithium Titane (LTO) are some of the cell employed, each offering distinct energy densities, safety profiles, and voltage limitations. The construction of these battery cells involves layering electrodes, electrolytes, and separators, with the total capacity determined by the quantity and dimensions of these layers.

- Cell types: Battery cells in EVs come in three primary shapes: cylindrical, prismatic, and pouch, each with distinct dimensions and designs, affecting how they're disassembled and recycled [44].

  1. Cylindrical cells, common in certain Tesla models, feature spirally wound electrodes, posing challenges for disassembly and direct recycling due to their compact structure and the use of epoxy resins, making them difficult to separate.
  2. Prismatic Cells, as those in the BMW i3, have a more squared shape and are enclosed in a rigid casing. They require special tools for opening and can be under significant pressure, posing a risk during the opening process.
  3. Pouch cells, used for example in the Nissan Leaf EVB, lack hard casings, easing opening and recycling efforts, though their material composition

may affect the economic feasibility of certain pyrometallurgy recycling process.

- Module size and power: Cell grouping into modules exhibits significant diversity in size and configuration across various car manufacturers and models.
- Cooling system: Effective thermal management is essential for mitigating rapid battery wear and aging. Manufacturers employ various cooling technologies, including forced air, liquid cooling, or natural refrigeration. Innovative approaches like phase-change materials and heat pipes are also being explored. Liquid cooling, in particular, offers diverse configurations, from simple under-battery cooling plates to more complex systems that cool individual cells.

Batteries are integrated into vehicles, adapting to the available space and thus contributing to the diversity of second-life solutions. In figure 1.4, batteries with the three different types of cells, and the resulting different solutions for modules, are shown. Battery dimensions and capacity differ based on the vehicle model. A Tesla Model S battery, for instance, might weigh around 530 kg and operate at 375 V, whereas a BMW i3's battery could be lighter at approximately 235 kg with a voltage of 355.2 V. Such variances not only influence the performance of the vehicle but also

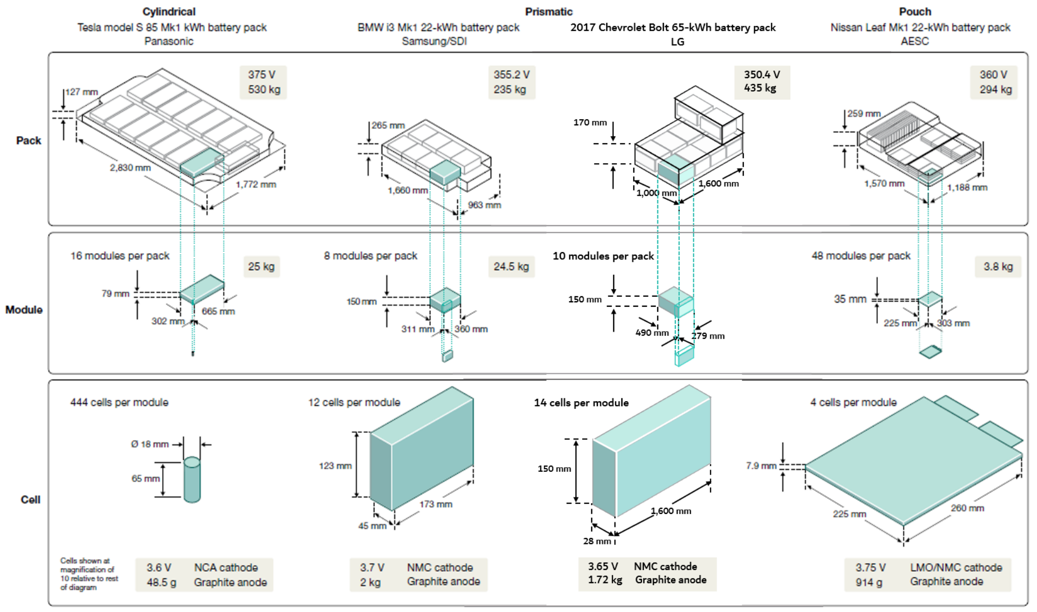


Fig. 1.4 Examples of batteries having the three different cell types, from [44].

pose unique challenges for recycling, given the diversity of materials and disassembly complexity. In [45] it is investigated a flexible gripper designed to automate the disassembly of lithium-ion batteries, addressing the challenges of handling diverse and potentially hazardous battery designs safely and efficiently. This enhances the recycling process by adapting to various battery geometries and chemistries, and reducing manual labor. As already stated, the first phase a battery undergoes after its first life is the disassembly, regardless it is reused, re-manufactured, reused in a stationary application, dismantled in cells or recycled. The work [46] investigates the application of stochastic optimization and artificial intelligence to refine the disassembly process of automotive parts, addressing the intrinsic uncertainties such as disassembly time and costs. Through a nuanced multiobjective optimization model, it showcases a method to balance the economic and environmental aspects of disassembly, illustrated with a detailed case study on automotive transmission disassembly. [47], instead, shifts the focus to the economic viability of disassembling EV Li-ion batteries, meticulously analyzing the costs associated with disassembly from the battery pack down to cells. The works illuminate the intricate balance between technical feasibility, economic viability, and environmental sustainability in recycling and reusing EV batteries and auto parts. In [48] a method for optimizing the disassembly sequence of EVBs to enhance the efficiency and environmental impact in re-manufacturing. It leverages a frame-subgroup structure in combination with a genetic algorithm to identify the disassembly precedence and connection relationships between components within EVBs, so to simplify the disassembly sequence planning (DSP) problem, making it easier to manage. [49] proposes a model aimed at designing an optimal disassembly process for EoL EV battery packs. The model seeks to achieve the highest economic profitability and leverages DSP to determine the optimal disassembly level and the most appropriate actions for the disassembled components; the model presented is applied to the lithium-ion battery from the Audi A3 Sportback e-tron Hybrid. The study [50], instead, underscores the pivotal role of artificial intelligence (AI) and machine learning (ML) in revolutionizing the automated disassembly of EV lithium-ion batteries (LIBs). It tackles the complexities and variability inherent in EV-LIBs by proposing AI and ML-driven solutions to streamline the disassembly process, thereby enhancing recycling efficiency and sustainability. In the same vein, considering the various solutions proposed for the disassembly, the work [51] proposes fully automated robotic disassembly of EV battery packs and proposes accurate identification of LIB components, generation of efficient disassembly plans, and precise robotic execution integrating computer vision and cognitive robotics. Also in [52] the introduction of AI for disassembly is investigated. Key components of the system considered include a Robot Cognition Processor for decision-making, a System Perception Unit for identifying and assessing parts, and a Disassembly Execution Unit for operational execution; the possibility to introduce collaborative robotics is investigated, too. In particular, [53] explores an innovative human-robot collaboration (HRC) framework, aimimg at optimizing the disassembly of retired EV batteries. It addresses the challenges posed by the variability and complexity of EV battery packs, such as diverse designs and materials, combining the dexterity and cognitive abilities of humans with the precision and strength of robots, and using depth vision systems and neural networks for precise identification and location of battery components. It results an improved efficiency, reducing labor intensity. Also Wegener, in [54, 55], analyzes the disassembly of lithium-ion EV batteries, focusing on those of the Audi Q5 Hybrid, to optimize recycling processes through detailed disassembly analysis and introduces collaborative robotics in a hybrid human-robot workstation for EOL EV battery disassembly, highlighting the role of robots in performing repetitive tasks and the importance of autonomous tool changing for efficiency and safety. Collaborative robotics is suggested also in [56], where a theoretical workstation that combines automated robotic arms and specialized tools to minimize manual labor and enhance safety is presented. Key innovations include a robotic unscrewing device, a flexible gripper, and a separation tool for dismantling battery components. In [57], identifying manual disassembly as costly and hazardous, robotic solution is presented for safer and faster operations. As experiments show that robots excel in cutting but are slower in pick-and-place tasks, a hybrid approach with human-robot collaboration for optimal efficiency and safety is proposed. [58] underscores the variety in BEV battery pack designs among manufacturers, stressing the necessity for an in-depth analysis of shared design elements to facilitate automation. It introduces a conceptual framework for an automated disassembly cell, specifically designed to accommodate these design variations. Central to this approach is the integration of collaborative robotics, armed with advanced safety mechanisms, to streamline the disassembly process. By delving into the unique challenges presented by various cell chemistries and battery configurations, a sophisticated, robot-assisted strategy is advocated, improving the efficiency of recycling operations.

The studies reported highlight the potentiality of collaborative robotics for the disassembly phase of EVB recycling. Therefore, this work will concentrate on the development of a collaborative cell for the disassembly of electric vehicle batteries.

1.3 Collaborative robotics

Human-robot collaboration (HRC) research is advancing rapidly, with recent studies focusing on enhancing interaction and collaboration between humans and robots, in various domains, including manufacturing, healthcare, and daily life. HRC explores the integration of robots into human activities, creating a tightly coupled dynamic system where humans, robots, and the environment interact to accomplish tasks. This collaboration is not limited to physical contact, but encompasses commitment in time and space, that needs mental and physical coordination. The development of collaborative robotics within industrial settings aims to create safe robotic systems that allow for the removal of barriers between robots and human operators. This approach enables humans and robots to work together seamlessly, as highlighted in early studies introducing the concept of collaborative robots, or cobots: mechanically compliant devices designed to assist workers. Today's collaborative robots are equipped with features to ensure safety and facilitate interaction, such as lightweight materials, rounded edges, and advanced sensors for managing undesired contacts. The use of collaborative robots allows a paradigm shift in the design and implementation of robotic cells, facilitating a synergistic workspace where humans and robots can operate in concert. This integration wants to exploit the unique strengths of both entities: the precision and repeatability inherent to robotic systems, and the flexibility and cognitive capabilities characteristic of human operators. Such a collaborative environment will enables the execution of tasks of heightened complexity. This contributes to the development of novel solutions in solving complex problems and in the execution of processes within the field of robotics. In the domain of industrial automation, HRC is delineated as collaborative robotics. This paradigm is based on the development of robotic systems engineered for safety, thereby removing the need for physical barriers traditionally demarcating the operational domains of robots from those of human operators. Consequently, this framework allows for the coalescence of human and robotic efforts in the execution of tasks. Cobots integrate both active and passive security features: they are designed with round surfaces and with light materials, and are provided with software that allow to stop the robots when contact forces or joint torques exceed defined thresholds. In figure 1.5, three examples of collaborative robotic arms are shown.

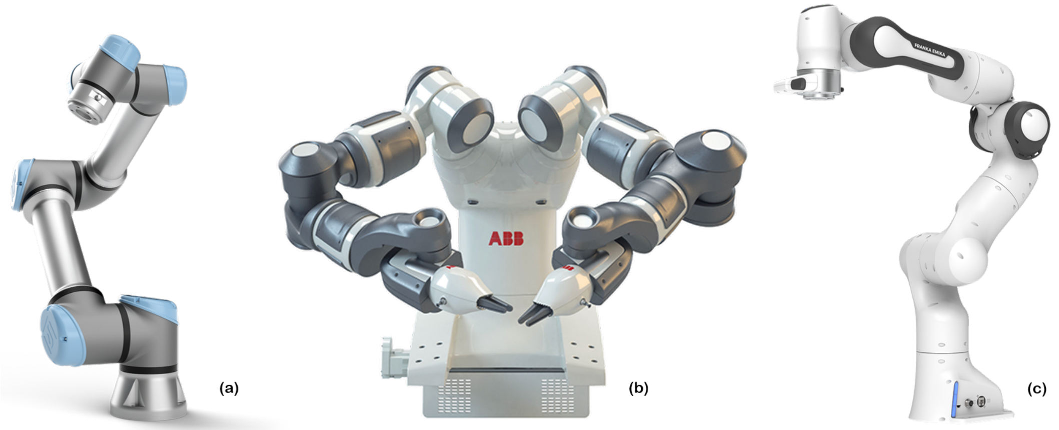


Fig. 1.5 Three examples of collaborative robotic arms. (a) is UR5 e-series from Universal Robots; (b) is Dual-arm YuMi® - IRB 14000 from ABB; (c) is Franka Research 3 from Franka Robotics.

However, the integration of the safety features does not ensure a safe collaboration between robot and operator, as the working at small distance with respect to the robot exposes the operator to possible danger. The International Federation of Robotics (IFR) [59] has delineated a distinction of the level of interaction between humans and robots, that is contingent upon the nature of their respective activities. The levels of collaborations can be distinguished as:

- Cell: the robot is closed in fences, and no cooperation exists between robot and worker.
- Coexistence: the fence is removed, but worker and robot do not share the workspace.
- Sequential collaboration: the robot and the worker share the workspace, but the movements of the two are sequential.
- Cooperation: the robot and the operator work at the same time on the same part, and they are both in motion.
- Responsive collaboration: the robot responds in real-time to the actions of the operator.

The types of collaborations are depicted in figure 1.6. Currently, the most used collaboration strategies are the coexistence and the sequential collaboration. However, the type of collaboration that would allow to better exploit the potentiality of HRC, and at the same time the most challenging, is the responsive collaboration. It becomes necessary to find strategies that ensure safe HRC. Two distinct methodologies are available to ensure safety within collaborative robotics: Power and Force Limiting (PFL) and Speed and Separation Monitoring (SSM). The PFL approach integrates comprehensive risk evaluation with mitigative strategies, ensuring that any contact between the operator and the collaborative robot is non-injurious. This entails not only employing passive safety designs, but also proactively managing the energy of the robot components by confining forces, torques, speeds, or momenta to predetermined thresholds. Conversely, SSM prioritizes the maintenance of a safe distance between the robot and the operator, without imposing direct restrictions on the robot's operational parameters. Operationally, this means that when an operator enters the cobot working area, the system dynamically adjusts its operations, devising and executing alternative paths that avoid collisions. This necessitates the employment of human detection technologies to surveil the proximal separation and the integration of collision avoidance mechanisms that can instantaneously revise the cobot path. In this work, the SSM strategy is implemented. The human tracking can be attained using wearable systems or by vision systems. The robot, the operator, the tracking system and the collision avoidance strategy are the elements that constitute the collaborative system.

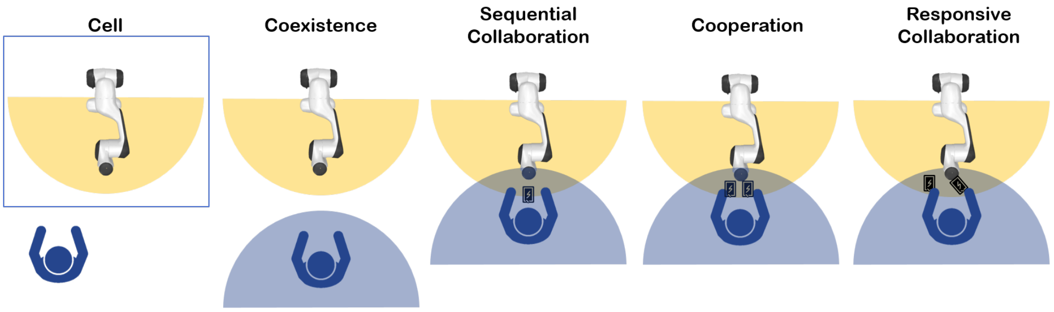


Fig. 1.6 Types of Human-Industrial Robot Collaboration, according to [59].

1.4 Work description

The present work investigates the problem of dismantling and recycling electric and electronic dismissed systems, with a special focus on exhausted batteries of electric vehicles. Their recycling process involves two main phases: a first one during which the main components of the battery pack are disassembled and a second one during which the cells are sent to a final plant for row material recycling. Electric vehicle batteries can benefit from collaborative disassembling, as the process requires flexibility due to the lack of standard design, that makes difficult to define a universal disassembly strategy. Furthermore, the complexity of disassembling electric vehicle batteries necessitates the involvement of a human operator to oversee and manage operations effectively. Concurrently, the prevalence of repetitive tasks naturally lends to the integration of robotic assistance. Starting from these assumptions, the work proposes a collaborative robotic cell. It comprises the human operator, a mobile manipulator, that consists of a collaborative robotic arm mounted on a mobile robot, and a safety strategy standing on 3D cameras and collision avoidance algorithms. The study delves into the dynamic interplay between humans and robots within shared workspaces. A special attention is devoted to safety aspects in the cooperation between man and robot. The work begins with a comprehensive analytical approach aimed at identifying the optimal sequence for disassembling batteries. This preliminary step leads to the strategic allocation and sequencing of tasks. This approach underscores the critical role of human-robot collaboration in enhancing both efficiency and safety in manufacturing processes.

To provide a cohesive structure, the thesis is organized into distinct chapters that address the problem from both high-level and detailed perspectives. Initially, the high-level challenges of disassembling large electric waste products, such as electric vehicle batteries, are outlined. This includes the identification of key tasks required for complete disassembly. Subsequently, the focus shifts to a detailed examination of introducing a mobile robot into the disassembly process, developing the mechanical model of the mobile robot and formulating control strategies to ensure precise and efficient operation.

Then, the type of robots are selected and a dynamic model of the robotic system is developed, identified and validated. Central to the development of the work are simulations of key scenarios that demonstrate the effective collaboration during the disassembly phase. The integration of advanced collision avoidance algorithms within the robot trajectory planning enable precise navigation and optimal trajectory definition according to human operator preferences, to enhance safety. The collision avoidance strategies are applied both to the mobile robot and to the robotic arm. The work also introduces a vision system for real-case scenarios, employing a body-tracking camera to provide real-time operator movement data, which assists in avoiding collisions and ensuring fluid human-robot interactions. In the end, the versatile robotic cell presented is a good example for both assembling and disassembling large, complex components, showcasing the broad applicability and potential of collaborative robotics in the manufacturing sector.

The robotic system model was developed using a MiR250 mobile robot and UR5 anthropomorphic arm, which were disposable. Subsequently, collaborative robotics algorithms on a mobile robot were tested on a Turtlebot3, which is easy to handle, requires minimal space for operation, and is ideal for quickly learning robotic programming with Robot Operating System (ROS). Finally, once the collaborative robotic cell was fully defined, the Franka FR3 and PAL ROBOTICS Tiago robots were selected due to their specific features, detailed in chapter 5. The model of the MiR250, which is the focus of the robotic modeling, is easily transferable to the Tiago robot as they both use differential driving wheels.

The thesis presents a dual perspective: it first provides a high-level framework for understanding the disassembly of electric vehicle batteries and then offers a detailed, practical implementation of a collaborative robotic system. This harmonized approach addresses the complexities of electronic waste disassembly. The work investigates the application of collaborative robotics in the initial disassembly process of electric vehicle batteries, considering the case study of the 2017 Chevrolet Bolt battery.

Chapter 2

Electric vehicle batteries

As already seen in section 1.2.2, the disassembly of the EVB is a compulsory step for the second life o recycling of the battery itself, whatever the preferrable EoL solution is. In this chapter, the description of the EV battery selected as case study is presented; then the disassembly process is examined and disassembly solutions are studied. In the end, the integration of the robotic system is analysed, highlighting the advantages it can lead, and the development of the work is presented.

2.1 EV battery description

The dismantling process of batteries presents a series of significant challenges that remain impediments to finding an ideal solution. Among these issues, the variability in battery design stands out. As standards for batteries are still in the process of being established, there is no overarching regulation requiring manufacturers to conform to a uniform design criterion. This results in batteries being tailor-made to fit the specifications of individual vehicles. Consequently, this bespoke approach to battery design has led to a wide array of models. While these models may share similar components, differences in their configuration and dimensions necessitate unique disassembly processes for each type. Despite the diversity in battery designs, the presence of common components across all batteries, such as battery modules, electrical connections, electrical components, and brackets) enables the formulation of a preliminary sequence for their removal and disassembly [54]:

- removal of the cover;
- unplug of the battery modules from other electrical components;
- mechanical disconnection of the battery modules and electrical components from the battery base;
- extraction of the electrical components;
- extraction of the battery modules;
- disassembly of the individual battery modules.

Given the extensive variety of batteries, this study has elected to focus on the case of the 2017 Chevrolet Bolt battery. This selection is grounded in the abundance of information available both in the literature, especially in [60], where main information is presented, and online regarding this specific model, thereby facilitating a comprehensive study despite the physical absence of the battery itself. This approach enables an in-depth examination of the disassembly process, potential recycling methodologies, and the evaluation of safety protocols without the need for direct interaction with the physical battery. The figure 2.1 shows the battery in question, without the upper cover. The battery has dimensions of about 1600x1000x200 mm.

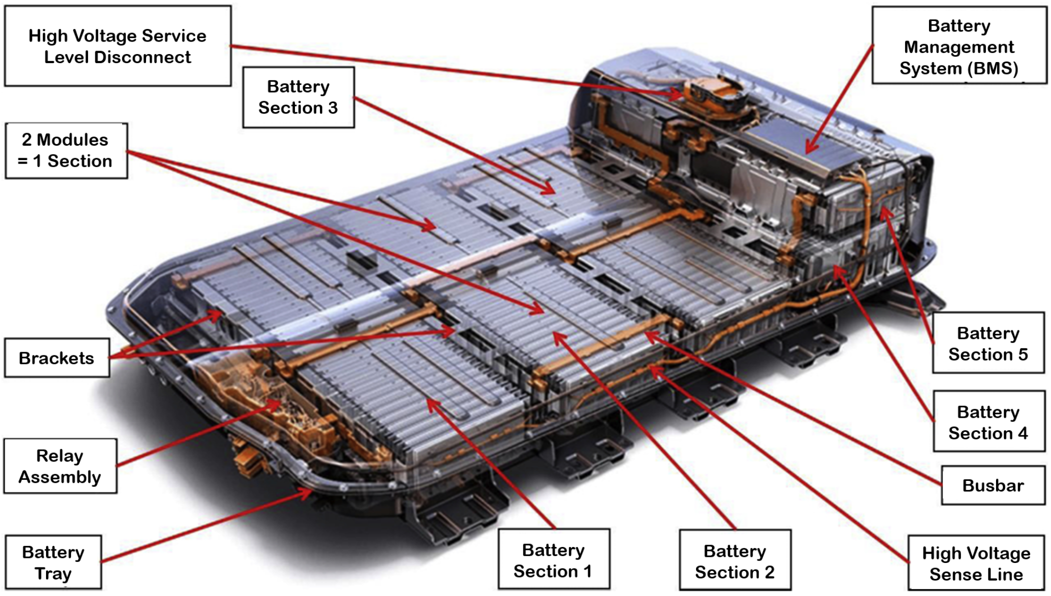


Fig. 2.1 2017 Chevrolet Bolt battery, with the main elements shown. From [60].

As can be seen, the battery presents five sections, made up of two battery modules each. A methodology has been defined to extract a feasible disassembly sequence. This involves the comprehensive identification of the battery constituents, which have been methodically classified into two distinct categories: parts (labeled with the letter 'P') and fastening elements (indicated by the letter 'F'). According to [61], 76 parts and 374 fastening elements were identified and labeled. The main parts of the battery are:

- battery modules;
- busbars;
- mechanical connections;
- Battery Management System (BMS);
- electronics;

while the fastening elements are:

- nuts;
- bolts;
- screws.

Of course, the disassembly sequence depends on the priority for the disassembly each components has.

2.2 Disassembly process

In the figures 2.2, 2.3 and 2.4 the disassembly graph of the Chevrolet Bolt 2017 EVB is shown [60]. The fasteners are in blue boxes, the parts in gray boxes, while in the orange circles the steps connecting with the other parts of the graph are highlighted. The graph was established by evaluating, for each element of the battery, which needs to be dismantled prior to others. The elements that can be disassembled together are grouped together in the graph, in order to reduce the disassembly steps and simplify the graph.

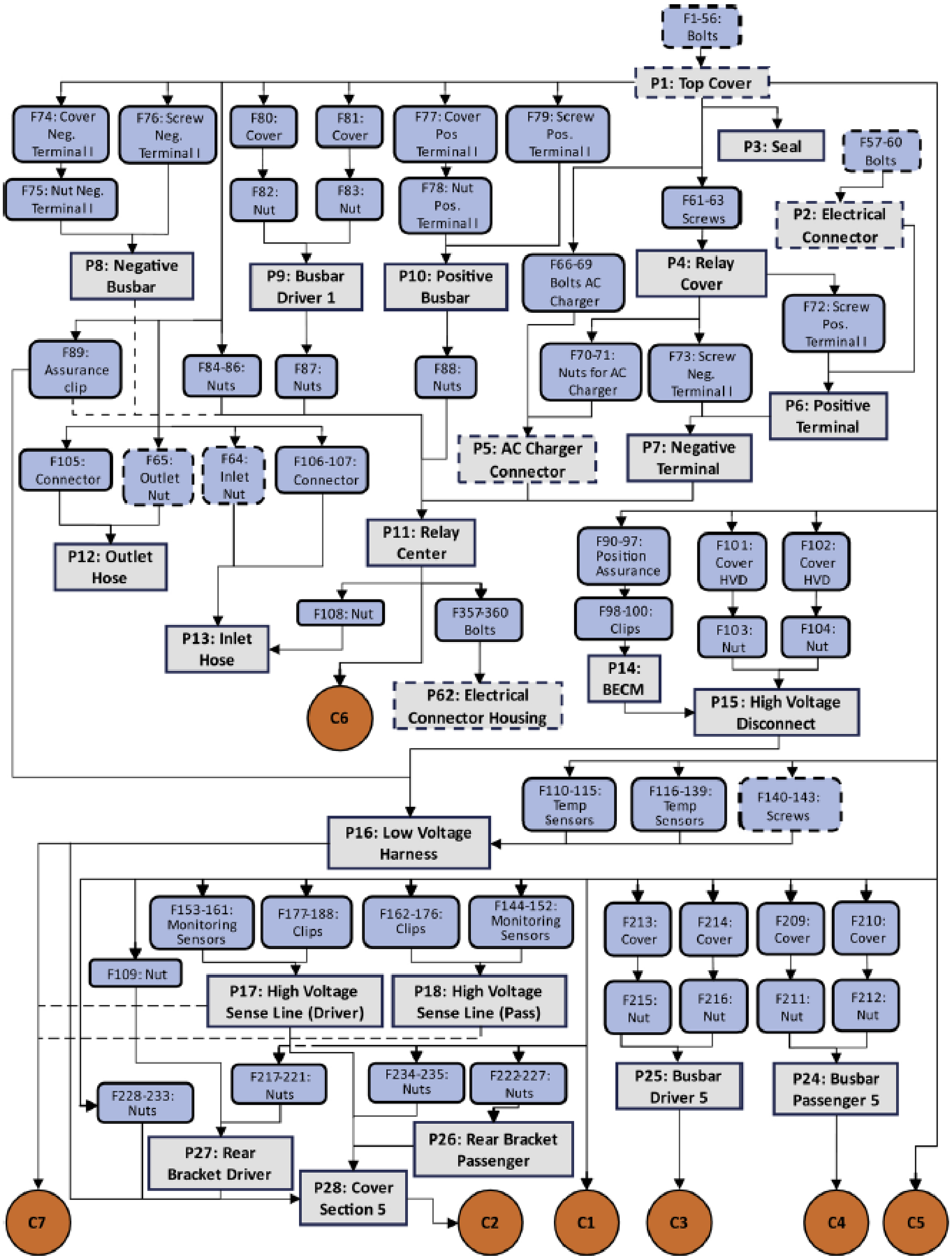


Fig. 2.2 Disassembly graph of Chevrolet Bolt 2017 EVB, part 1 [60].

A method for determining if each disassembly step, or task, can be automated is used. It consists in defining two main key indicators, the technical ability of a disassembly process to be automated (TAA) and the necessity to automate the corresponding disassembly operation (NA) [62]. According to [60], five TAA and five NA are defined. They are presented in table 2.1 For each task, a value for the

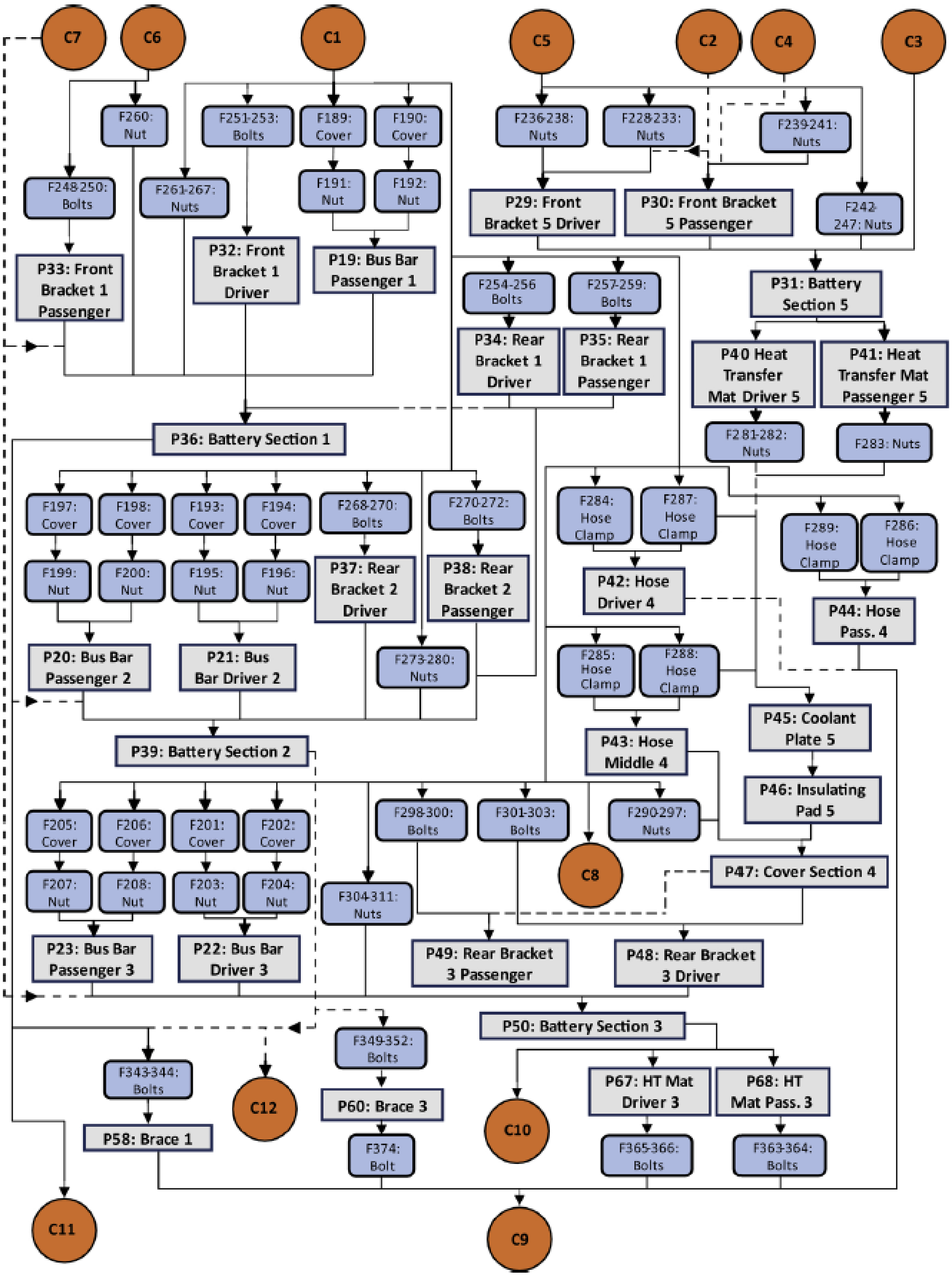


Fig. 2.3 Disassembly graph of Chevrolet Bolt 2017 EVB, part 2 [60].

TAAs and NAs is attributed. These scores determine if each task can be automated. Each criteria can have a score that goes from -2 to +2, and has a weighting factor equal to 10. A task can be considered for automation if it has a high score of NA and TAA. For each task the values of the NA and of the TAA have been calculated and plotted on a graph, that has TAA on the x-axis and NA on the y-axis. The higher the value of NA, the higher is the necessity to automate that disassembly step. The higher the value of TAA, the higher is the possibility to automate that step. The resulting plot of the Chevrolet Bolt 2017 EVB is presented in figure 2.5. Each blue

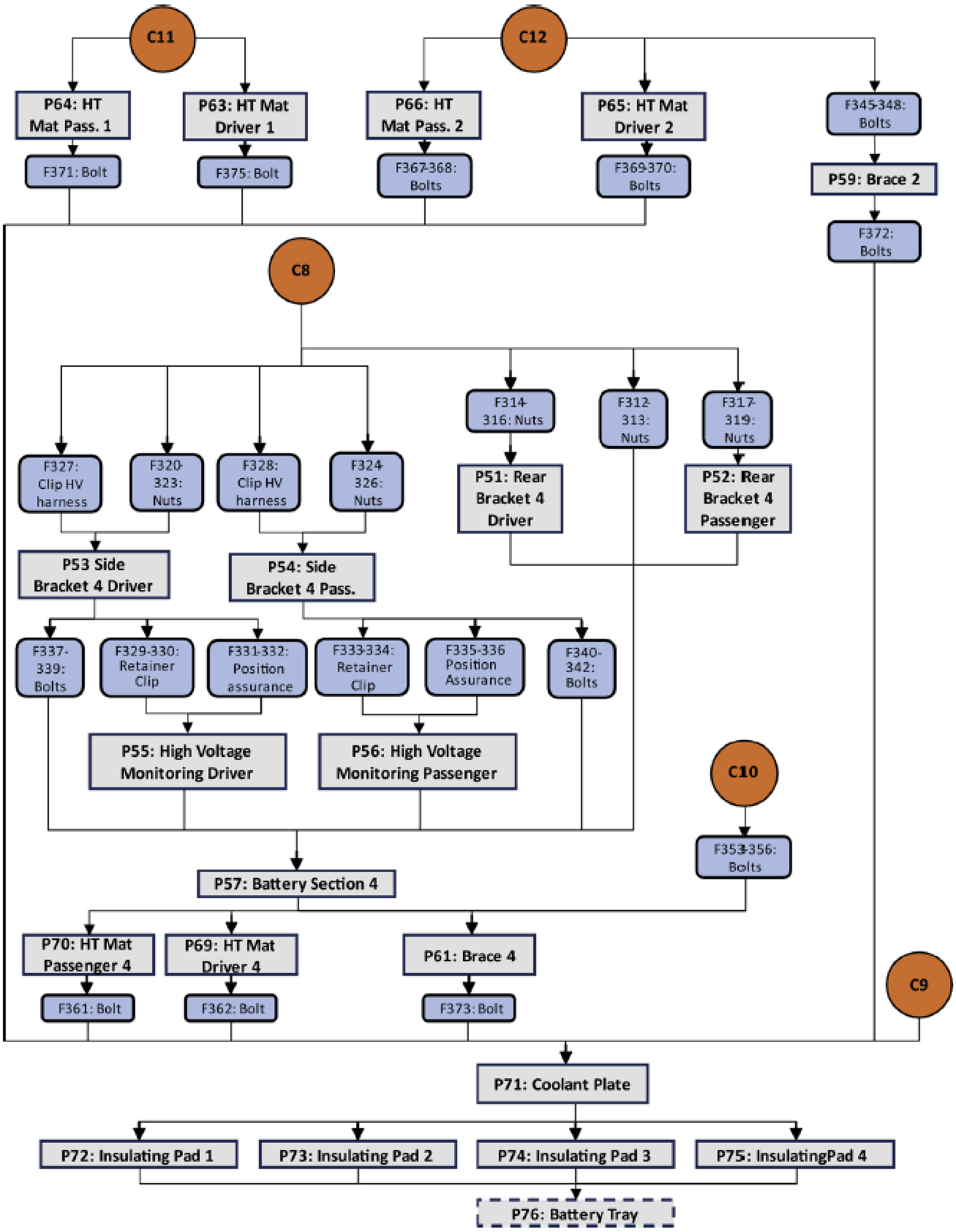


Fig. 2.4 Disassembly graph of Chevrolet Bolt 2017 EVB, part 3 [60].

point in the chart represent a disassembly step. In the green shaded area fall the disassembly tasks that should be automated.
In table 2.2 the values of each step for the Chevrolet Bolt 2017 EVB are reported. It is important to underline that each step in the table involves more different steps of the same type and that can be done with the same tool.

Table 2.1 NA and TAA criterions, [60]

Category	Criterion #	Criterion Description
NA	1	Number of Motions (human)
	2	Duration of manual disassembly time in seconds
	3	Danger (High voltage protection, hazardous materials)
	4	Weight
	5	Priority (value)
TAA	1	Complexity of motion (for robot, number of different motions)
	2	Access for end effector
	3	Possible detection
	4	Automation potential for robotic end effector
	5	Material handling

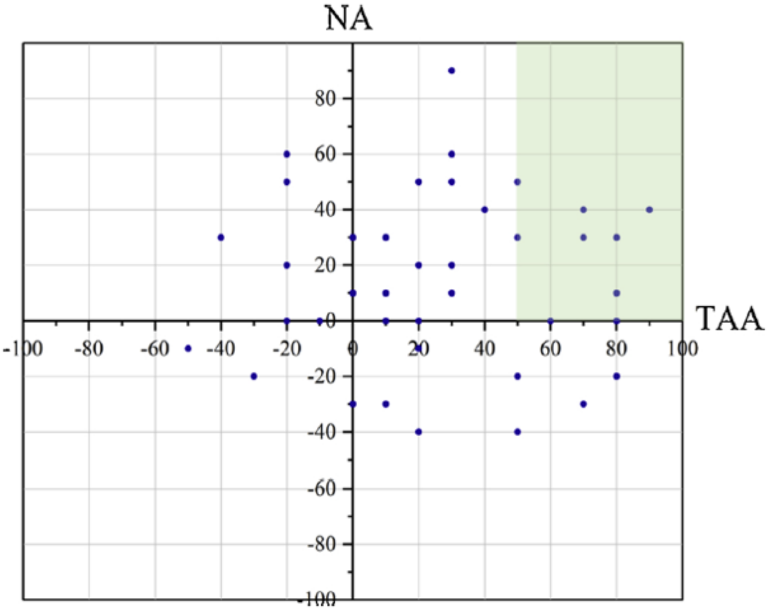


Fig. 2.5 NA and TAA for the tasks in the disassembly of the Chevrolet Bolt 2017, [60].

Table 2.2 Disassembly tasks of the Chevrolet Bolt 2017 BEV, [60]

Step	Involved Parts or Fasteners	Description	NA	TAA
D1	F1-56	Bolts for Top Cover	40	90
D2	P1	Lifting of Top Cover	10	30
D3	F57-60	Bolts for Electrical Connector	30	70
D4	P2	Electrical Connector	-20	50
D5	P3	Seal	-30	0
D6	F74, F77, F80, F81	Covers for Busbars (Front)	20	20
D7	F61-63, F66-69, F75-76, F78-79, F82-83	Nuts/ Screws for Busbars (Front) and Relay Cover	60	30
D8	P4, P8-10	Relay Cover, Busbars	30	50
D9	F70-73, F84-88	Nuts, Screws below Relay Cover	50	30
D10	P5-7, P11	Grabbing of Relay Center, Terminals, AC-Charger	30	-40
D11	F105-108	Connectors and Nut for Coolant Hoses	-10	-50
D12	F64-65	Big Nuts for Coolant Hoses	-20	-30
D13	P12, P13	Grabbing of Coolant Hoses	-40	50
D14	F357-360	Nuts for Electrical Connector	0	60
D15	P62	Electrical Connector	-40	20
D16	F89-100, F110-139	Assurance Clips, Temp. Sensors, BMS clips	60	-20
D17	P14	BMS	0	80
D18	F101-102, F189-190, F193-194, F197-198, F201-202, F205-206 F209-210, F213-214	Covers for Busbars, High Voltage Disconnect	50	20
D19	F103-104, F109, F140-143, F191-192, F195-196, F199-200, F203-204, F207-208, F211-212, F215-280, F290-326	Nuts, Bolts and Screws for Busbars and Brackets, High Voltage Disconnect and Battery Sections	50	50

D20	P15	High Voltage Disconnect	0	60
D21	P16	Low Voltage Harness	0	-20
D22	F144-188	Clips and Temp. Sensors for HV Sense Lines	50	-20
D23	P17, P18	HV Sense Lines	20	-20
D24	P26-27	Rear Brackets	0	60
D25	P28	Cover Battery Section 5	20	30
D26	P19-25, P29-30, P32-35, P37-38, P51-52, P53-54	Busbars and Brackets	50	30
D27	P40-41	HT Mats Battery Section 5	10	10
D28	F284-289	Hose Champs	10	0
D29	P42-44	Hoses	-30	10
D30	F281-283	Nuts for Coolant Plate	30	80
D31	P45	Coolant Plate Section 5	30	10
D32	P46	Insulating Pad Battery Section 5	0	20
D33	P47	Cover Battery Section 4	20	30
D34	F327-328	Clips HV Harness Battery Section 4	0	-10
D35	P48-49	Side Brackets Battery Section 4	0	60
D36	F337-342	Bolts Battery Section 4	40	70
D37	F329-336	Retainers, Position Assurance Battery Section 4	30	0
D38	P55-56	HV Monitoring Circuit Battery Section 4	0	60
D39	P36, P39, P50, P57	Battery Sections 1–4 Lifting	90	30
D40	P63-70	Heat Transfer Mats	10	10
D41	F343-356, F361-371, F375	Bolts for Braces, Coolant Plate	10	80
D42	P58-61	Braces	-30	70
D43	F372-374	Bolts for Coolant Plate	-20	80
D44	P71	Coolant Plate	0	10
D45	P72-75	Insulating Pads	-10	20
D46	P76	Handling of Battery Tray	40	40

Once all the steps are analysed, and the possible tasks to be automited are detected, it becomes necessary to define possible disassembly sequences for the EV battery.

2.3 Definition of possible disassembly sequences

Once the disassembly process is defined, determining a disassembly sequence becomes mandatory. The disassembly sequence dictates the total disassembly time and the interaction between the robot and the human operator. Clearly, the optimal solution would be the one with the shortest time. The issue is not as trivial as it might seem. If the disassembly were to be conducted by a human operator alone, the variation in disassembly time would be minimal, barring tool change time and the time required for moving from one part of the battery to another to perform various tasks. Essentially, it would primarily be determined by the sum of the times needed to execute each individual task. The only real constraint would be ensuring that each task can be performed once all the prerequisite tasks have been completed. Now, instead, the disassembly sequence has to be allocated between robot and human operator. In this work, only one worker and one robot are considered.

In addressing the problem at hand, it is necessary to allocate specific tasks to the human operator and others to the robot. As indicated in 2.2, although there are tasks that should preferably be executed by the robot and others by the human, tasks are categorized into those that can only be performed by the human, due to the robot's inability to carry out such operations, and tasks that can be executed by both the human and the robot. This distinction allows for a more efficient division of labor, ensuring that operations are executed in the most time-efficient manner possible. There may be instances where, within a given time frame, the human operator and the robot are required to perform different tasks in close proximity to each other. This would lead to the stop of the robot, in order to allow the operator to work safely. Given these premises, it is imperative to delineate the specific tasks to be undertaken by the human operator and those to be performed by the robot.

An algorithm has been developed to establish the disassembly sequences, with the disassembly cycle time defined as the greater of the cycle times between the human operator and the robot, ensuring synchronization and efficiency in the collaborative task execution. For defining the algorithm, the following features have been considered:

- the disassembly process and hierarchy;
- the position of the diffent parts in the battery;
- the workspace of the task areas, according to the operator that performs the task;
- the cycle time of a task needed by the operator;
- the time necessary to move to one part to another of the working area;
- the changing tool time;
- the stop/deceleration time of the operators due to the sharing of the working area;
- the idle time.

The following simplification assumptions have been done:

- the cycle time of the tasks have been supposed;
- the tool changing times have been considered constant;
- the two operators can work in the same working area.

A code has been assigned to all the tasks, in order to simplify the final table. Then, an excel file has been produced. It contains all the information about the tasks:

- the name of the task;
- the position of the components to dismantle on the battery;
- the operator that performs the task;
- the tools necessary for the task;
- the cycle time for the task;

- the tasks that have to be performed before the reference task.

The algorithm returns the disassembly sequence for robot and human operator, the total time for disassembly of the robot and of the human operator, the idle times of robot and human, changing tool time and the time for the single tasks, the stop time due to the sharing of the workspace between robot and human operator.
The algorithm reads the data from the Excel file and stores them in appropriate variables. Then, the it operates within a while loop that includes a series of conditional statements that analyze the various possible scenarios that can arise. Starting from a total number N_{tot} of tasks to be performed, when considering the j^{th} cycle, there remains a number $N_{tot} - j$ of tasks still to be executed. These can be divided into: tasks that can be performed by the human operator and tasks that can be performed by either human or robot. Additionally, tasks must be further classified into two categories: tasks that can be performed immediately and tasks that cannot be performed yet. As an initial attempt, the strategy involved mapping out all conceivable disassembly sequence combinations. Starting from the initial tasks that could be performed, different tasks were then assigned to the operators, progressively defining an increasing number of disassembly sequences. This method allows for the eventual selection of the sequence with the shortest cycle time, optimizing the overall efficiency of the process. However, it was observed that the number of potential disassembly sequences were excessively high, to such an extent that even advanced computational tools like Matlab struggled to process the data within an acceptable timeframe.
Consequently, a second method was adopted. Under this approach, when multiple tasks are eligible for execution, task selection is randomized by the algorithm. This modification ensures that the output is a singular disassembly sequence allocated between the human operator and the robot. Therefore, the algorithm is executed multiple times, allowing for the identification and selection of the sequence with the minimal cycle time. This iterative process enhances the practicality of the algorithm by reducing computational overload and focusing on optimizing operational efficiency. In figure 2.6 the results of the algorithm, that returns 80 possible disassembly sequences, are presented. The plot presents the cycle time of the disassembly sequences determined. The resulting times are coherent with those presented in [63]. The disassembly sequences found give fundamental information about how the disassembly operation takes place. As also the position of the task is recorded, from the disassembly sequence is possible to determine if and where human and robot share

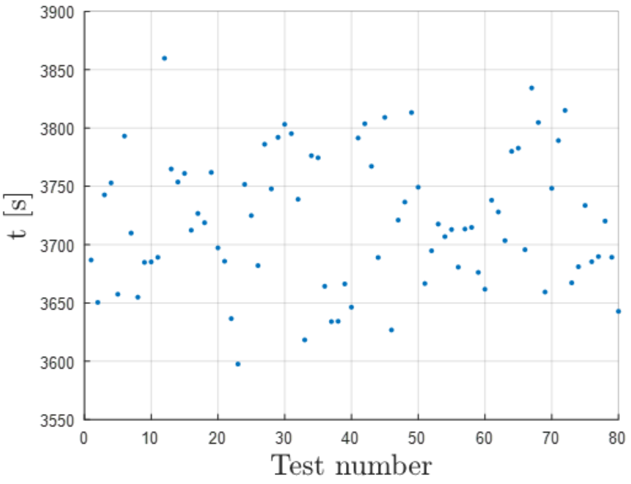


Fig. 2.6 Disassembly sequences time plot

workspace, and which are the activities they are performing. In particular, in figure 2.7 are presented the Gant diagrams of both robot and operator for a disassembly sequence. In different colors are presented the time when they share the workspace, in red, and the time when they do not share the workspace, in green. In white the time when one of them is forced to stop. It appears that there are many moments in which robot and operators need share the workspace. These results are common

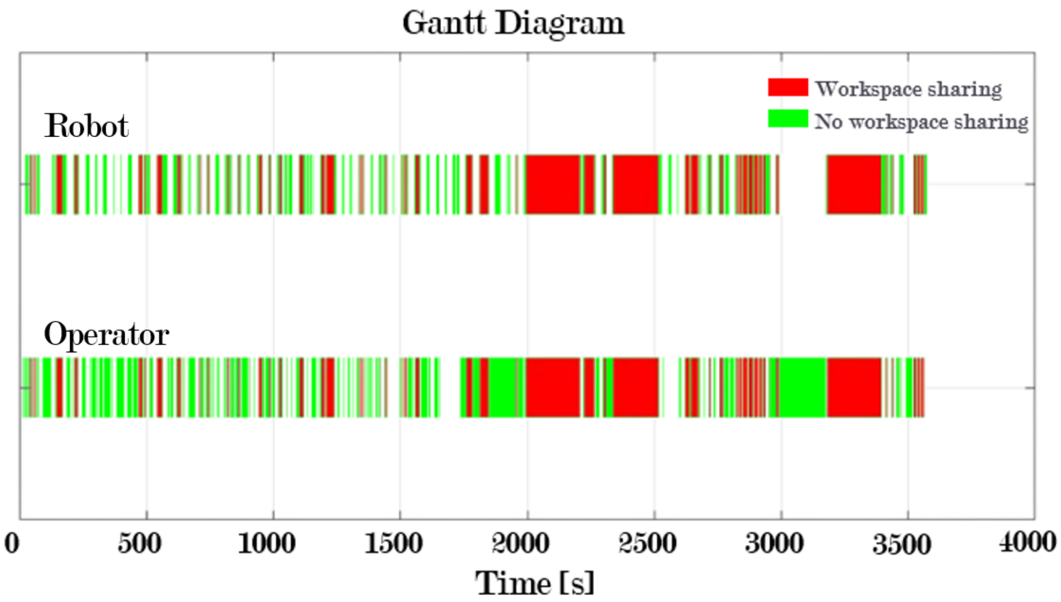


Fig. 2.7 Gant of a disassembly sequence, for robot and operator.

to all the disassembly sequences analyzed. This aspect will is analysed in the next section.

2.4 Robot integration

As anticipated, the disassembly of EVB, and in particular of the battery of a Chevrolet Bolt 2017, can benefit from the use of collaborative robotics. The prolonged shared workspace between human operators and robots underscores the necessity of implementing routines for safe space sharing within the system to facilitate disassembly tasks securely while simultaneously reducing instances in which the robot must halt operations, thereby minimizing downtime.

Furthermore, the dimensions of the battery significantly impact the configuration of the robotic cell required for disassembly.As discussed in the first chapter, the typical dimensions of electric vehicle batteries (EVBs) are quite large since they are positioned in the lower part of cars and serve a structural role, spanning the entire base of the vehicle. Consequently, it becomes evident that the workspaces of collaborative robots, which are generally smaller than those of traditional robots, are insufficient for handling such sizable components efficiently. Indeed, the variety in the dimensions of electric batteries, as investigated in chapter 1, complicates the integration of more robotic arms, as a highly rigid setup would result, limiting the flexibility required for efficient automation. For this reasons, the decision was made to utilize a mobile robot equipped with an anthropomorphic manipulator as the collaborative robotic system. This setup enhances flexibility and reach, accommodating the large dimensions of the tasks involved. This configuration enables the robotic system to serve multiple functions beyond disassembly, as transporting the dismantled materials to a designated area, autonomously navigating to a tool change station, and generally supporting various robotic activities that occur outside the collaborative cell.

With these considerations in place, the activities involving close proximity and potential interference between the human operator and the robot have been analysed. The number of tasks to perform is quite big. However, a common thread was identified among the various activities. Consequently, all potential interactions were categorized into three distinct groups:

1. the operator, changing its position relative to the EVB, needs to move behind the robot;
2. the robot, changing its position relative to the EVB, needs to move behind the operator;
3. the robot and the operator work in the same workspace of the battery.

The cases described are respectively represented in figures 2.8, 2.9 and 2.10 respectively.

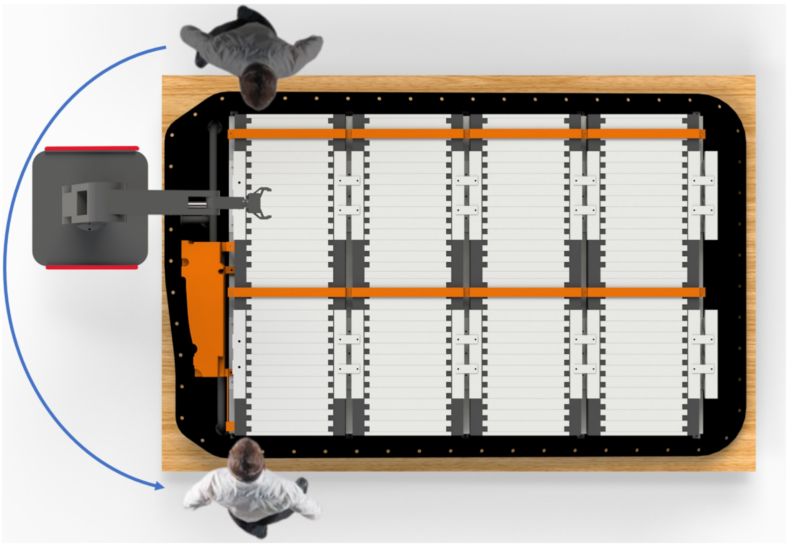


Fig. 2.8 Case 1: the operator, changing its position relative to the EVB, needs to move behind the robot.

2.5 Conclusions

The disassembly of a complex structure as an electric vehicle battery presents significant challenges. Given the substantial dimensions of these batteries, the use of a single robotic arm would bring limited benefits. That's why the solution identified involves a robotic manipulator, that consists of robotic arm mounted on a mobile robot, within the disassembly cell. In this sense, collaborative robotics becomes a crucial component of the activity proposed. It is essential, indeed, to assess and implement potential strategies to enhance the coordination between human

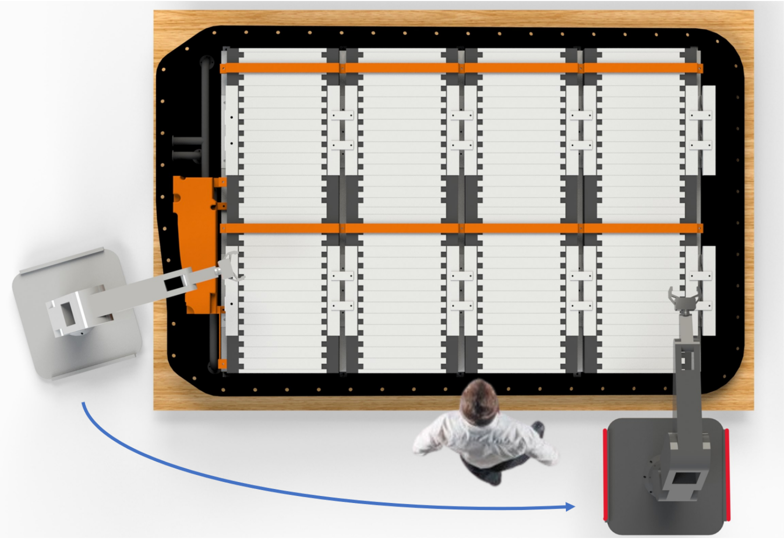


Fig. 2.9 Case 2: the robot, changing its position relative to the EVB, needs to move behind the operator.

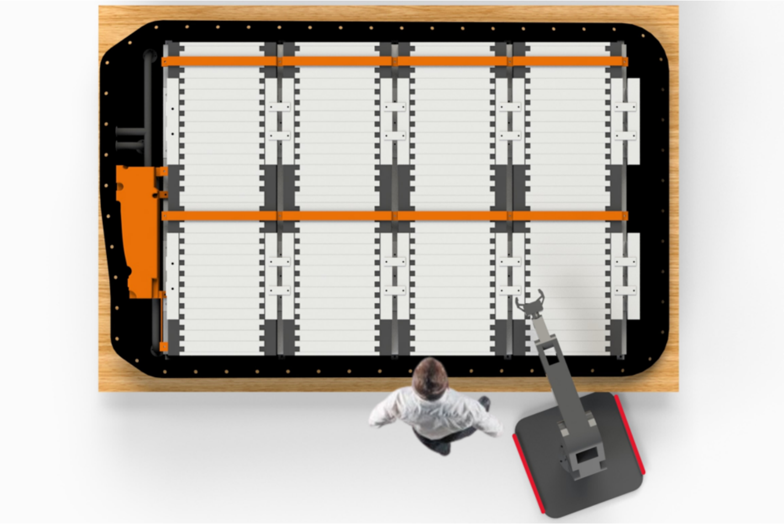


Fig. 2.10 Case 3: the robot and the operator work in the same workspace of the battery.

and robot, optimizing the efficiency and safety of the disassembly process. This approach ensures both the protection of human operators and the continuity of robotic functions, contributing to overall operational effectiveness.

Chapter 3

Robotic system dynamic model

Once established that the disassembly of electric vehicle batteries will be conducted using a robotic manipulator in collaboration with a human operator, this chapter delves into the subsequent development of a dynamic model of the robotic system. The decision to implement a collaborative robotic solution sets the stage for a detailed exploration of the dynamic model. This model serves as a pivotal tool for simulating and optimizing the behavior in an industrial environment, increasing the capability to monitor their performance to detect failures or deviation from expected behavior. In this chapter, the model, along with subsequent identification and validation, of a system comprising a UR5 mounted on a MiR250 is presented. For the validation of the system, a motion tracking systems that comprises four Flex13 cameras from Optitrack is used. The robots were chosen due to the availability of ample space for their operation and the feasibility of their utilization. However, for subsequent applications, these robots, as well as the camera system used for identification and validation, were no longer available. Consequently, it was not possible to further develop and validate the model with the definitive robotic system. Nonetheless, the constructed model and the validation method employed can be readily replicated on the final robotic system. This approach ensures that the foundational research can be effectively applied and tested in future implementations.

The model of a robotic arm is a topic that has been extensively addressed in the literature, whereas models of mobile robots with caster wheels, as the MiR250 is, have not been as thoroughly explored. Moreover, the model of the UR5 has been developed and validated by Raviola, in [64]. Consequently, the treatment of the

MiR250 model will be much more detailed, while the model of the UR5 is only mentioned.

3.1 Use of dynamic model

The integration of mobile robots with robotic arms represents a robust tool for factory automation, gaining increased popularity in recent years. These systems are adept at performing complex tasks including manipulation, assembly, inspection, and transportation across various industrial settings. Nonetheless, the design, testing, and control of such integrated systems present substantial challenges. This complexity stems from their multifaceted nature involving multiple components, sensors, actuators, and interactions with the environment.

[65] explores the challenges and requirements essential for the development and application of autonomous industrial mobile manipulator systems. Key focus areas include sustainability, system configuration, adaptation, autonomy, positioning, manipulation and grasping capabilities, robot-robot and human-robot interactions, process quality, dependability, and physical properties of the systems. Meanwhile, [66] details an autonomous mobile manipulator designed to adeptly handle system uncertainties and exceptions through coordinated control strategies that blend visual and force servoing with sophisticated reactive task control. Additionally, [67] presents a method for planning the coordinated motion of mobile manipulators that considers both stability and manipulation tasks, employing the concept of a valid stable region to assess stability amid disturbances.

Given the intricate nature of these systems, there is a pressing need for effective tools and methods to simulate, monitor, and optimize their behavior and performance. A promising solution is the adoption of digital twins—virtual models that replicate the physical systems states, properties, and dynamics in real-time and that exchange data with the physical system in order to provide a detailed knowledge of its status and evolution. This approach not only enhances understanding and management of mobile robot-arm systems but also fosters improvements in operational efficacy and safety. Digital twins enable semi-physical simulations that can significantly reduce the time and cost associated with physical commissioning and reconfiguration. By identifying design errors and flaws early in the development process, these virtual models help to minimize the financial impact associated with these issues [68, 69].

The implementation of dynamic models or digital twins allows production systems to compensate for the real-world uncertainties of the environment as well as the unpredictability of human behavior. Consequently, there is a reduced need for offline programming and validation of robot motions when changes in the process are necessary, thus saving time and effort [70]. Moreover, dynamic models facilitate the reduction of development costs and risks, enhance operational efficiency and quality, improve fault detection and diagnosis, and support decision-making and maintenance processes. However, developing accurate and reliable models for mobile robot-arm systems presents significant challenges. It requires modeling various aspects of the system across different levels of abstraction and fidelity, integrating heterogeneous data sources and formats, and validating the models against real-world measurements and observations. These complexities underscore the need for a sophisticated approach to ensure the effectiveness and reliability of dynamic models in industrial applications.

In this chapter, the model of a system consisting of a MiR250 mobile robot coupled with a UR5 anthropomorphic robot is presented. To enhance the model accuracy and reliability, parameter estimation and validation processes were conducted. These processes involved an experimental campaign using an Optitrack motion capture system to record the movements of the UR5 end effector and the MiR250 base center, ensuring that the dynamic model accurately reflects the physical behavior of the system. The expectation from the combination of robots is that their dynamics will interact with one another, thereby generating deviations in the trajectory planned for the MiR250 as well as that for the UR5. The objective is to identify these deviations by creating a dynamic model of the system that reproduces its real behavior as faithfully as possible.

3.2 Robot description

The robotic system employed for developing the dynamic model consists of a UR5 e-series collaborative robot and a MiR250 mobile robot. 3.1 displays an image of this integrated system. As depicted, a cabinet is positioned between the two robots, housing the control unit for the UR5. The resultant system integrates the pronounced manipulation and collaboration capabilities of a collaborative arm with

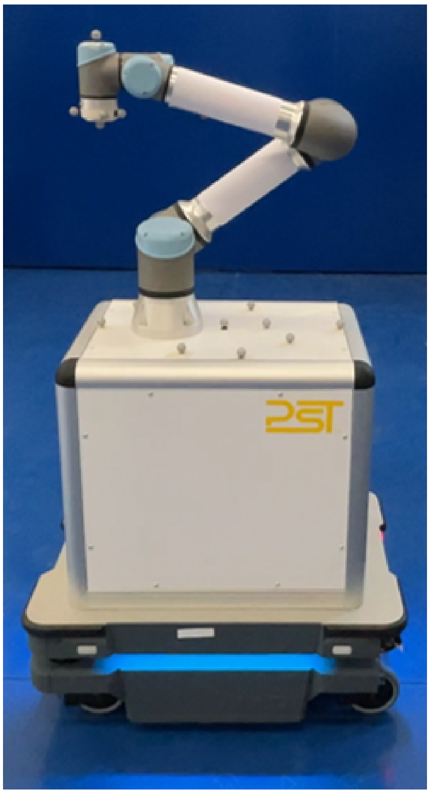


Fig. 3.1 Robotic system composed by MiR250 and UR5 e-series.

the autonomous mobility of a mobile robot, thereby achieving a high degree of versatility.

3.2.1 Mobile robot MiR250

The MiR250 is an Autonomous Mobile Robot (AMR), that is a subgroup of Automated Guided Vehicles (AGVs). While AGVs follow predefined paths using physical or virtual guides, AMRs represent a more advanced category with enhanced flexibility and autonomy. AGVs are programmed to follow fixed routes, typically using wires, magnetic strips, or markers embedded in the floor. They are best suited for environments where the routes do not need frequent changes, as modifying their path can be cumbersome and time-consuming. AGVs are often used for repetitive tasks over predictable paths and require a controlled environment to operate efficiently. On the other hand, AMRs are equipped with advanced sensors and on-board intelligence, enabling them to understand and navigate their environment dynamically. They do not require predefined paths and can autonomously calculate the best route to their destination, avoiding obstacles in real time. This flexibility allows AMRs to adapt quickly to changes in the environment, making them ideal for dynamic settings where obstacles and route requirements can frequently change. The MiR250 [71] is presented in figure 3.2. It is equipped with two SICK nanoscan3 laser scanners for localization and obstacle detection, two Intel RealSense D435 3D cameras, and eight proximity sensors. Additionally, it can utilize four digital inputs and four digital outputs. It can reach a maximum velocity of 2 m/s and has a payload of 250 kg. The positioning accuracy is 50 mm. The robot has two differential driving wheels and four caster wheels. In table 3.1 the technical specifications of the robot are reported.

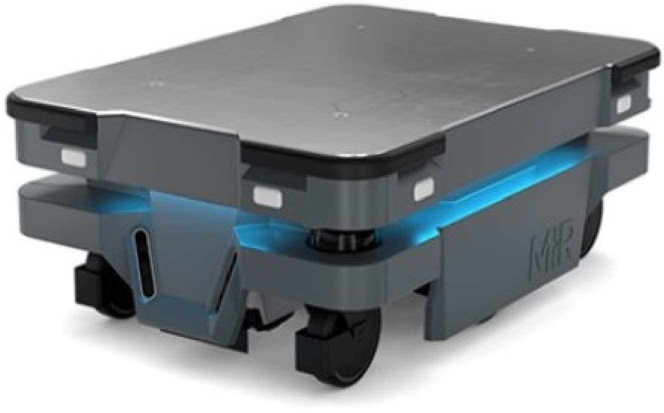


Fig. 3.2 Robot MiR250.

Table 3.1 Technical specification of MiR250.

Length	800 mm
Width	580 mm
Height	300 mm
Weight	83 kg
Driving wheel radius	100 mm
Caster wheel radius	62.5 mm
Driving wheel axis	410 mm
Caster wheel axis, on the width	375 mm
Caster wheel axis, on the length	500 mm
Reduction ratio	10.167

Caster wheels

Caster wheels are a type of wheel mounted on a swivel, which allows multi-directional movement. In figure 3.3, an example of caster wheel is shown. The key feature of casters is their ability to swivel around a vertical axis. This capability is enabled by a swivel bearing at the base of the mount, which allows the wheel to rotate 360°. This full range of motion permits the caster to move fluidly in any direction. As the attached object moves, in this case the mobile robot, the casters automatically align themselves in the direction of motion. This alignment optimizes the rolling path for reduced resistance and smoother movement. The caster wheels on the MiR feature a vertical swivel axis that is offset from the wheel center of rotation, complicating the modeling of the wheel.

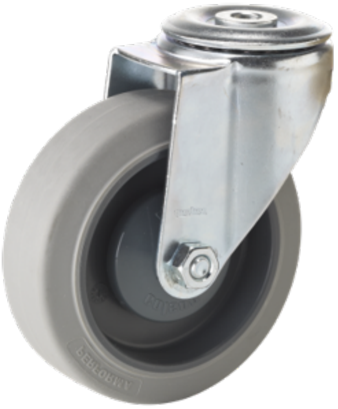


Fig. 3.3 Example of caster wheel.

Differential Wheeled Robot Kinematics

The mobile robot has three degrees of freedom: the position in the plane, defined as translation in $x-axis$, that allows the motion forward and backward, and as the translation in $y-axis$, for moving left or right, and the orientation, that is the rotation about the z-axis. The driving wheels do not steer, but their relative rotation allows the robot to change the direction of motion [72]. In figure 3.4 the top view of a rigid body consisting of two driving wheels and a frame that connects them to prevent relative motion along the direction $\overline{\eta}$ is shown. The reference frame $0-xyz$ of a mobile robot is oriented with $x-$axis in the driving direction, coinciding with $\overline{\tau}$,

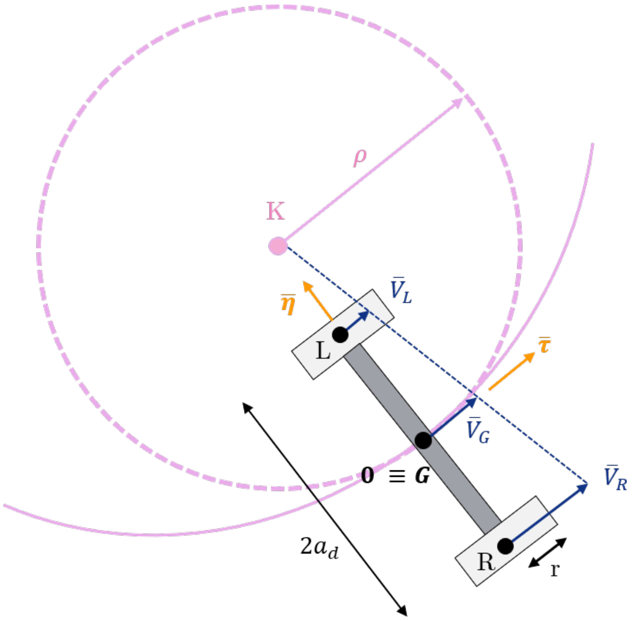


Fig. 3.4 Kinematic scheme of a differential wheeled robot.

$z-$axis on the vertical direction, pointing upward, and $y-$axis given by the right-hand rule, coinciding with $\overline{\eta}$. The center of the reference frame 0, that coincides with the center of mass G, is located in the center of the robot, as its structure is symmetric. $0-xyz$ is a mobile reference frame. The Jacobian matrix of the mobile robot J_{mr} that relates the velocity of the robot with the velocity of the driving wheels is:

$$\mathbf{J}_{mr} = \begin{bmatrix} r\cos\varphi & r\cos\varphi \\ r\sin\varphi & r\sin\varphi \\ r/(2a_a) & -r/(2a_d) \end{bmatrix} \tag{3.1}$$

in which φ is the orientation of the robot, r is the radius of the driving wheel and $2a_d$ is the axis of the driving wheels. It holds:

$$\begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{Bmatrix} = \mathbf{J}_{mr} \begin{Bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{Bmatrix} \tag{3.2}$$

where $\dot{X}$ is the robot velocity in X direction of the global reference frame, $\dot{Y}$ is the robot velocity in Y direction of the global reference frame, $\dot{\varphi}$ is the robot angular velocity in the global reference frame, and $\dot{\theta}_r$ and $\dot{\theta}_l$ are the angular velocities of the right and left wheel of the robot, respectively. On the contrary, it is possible to write:

$$\begin{Bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{Bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\varphi & \sin\varphi & 2a_d \\ \cos\varphi & \cos\varphi & -2a_d \end{bmatrix} \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{Bmatrix} \tag{3.3}$$

in which the central matrix is called generalized inverse of $\boldsymbol{J}_{mr}$, $\boldsymbol{J}_{mr}^{\dagger}$.

3.2.2 UR5 e-series

The UR5 e-series anthropomorphic arm from Universal Robots is a well-known 6-degree-of-freedom collaborative robot. It supports a payload capacity of 5 kg and boasts a reach of 850 mm. Each of its joints is capable of a full 360° rotation in both clockwise and counterclockwise directions, with a maximum velocity of $180°/s$. The robot repeatability is 0.03 mm. The weight of the robot is 20.6 kg. In figure 3.5 a picture of the robot is shown [73]. In table 3.2 the Denavit-Hartenberg (DH) according to modified convention of UR5 e-series are reported [74]. The values of θ_i, where i stands for the number of the robot joint, correspond to the joint angular values.

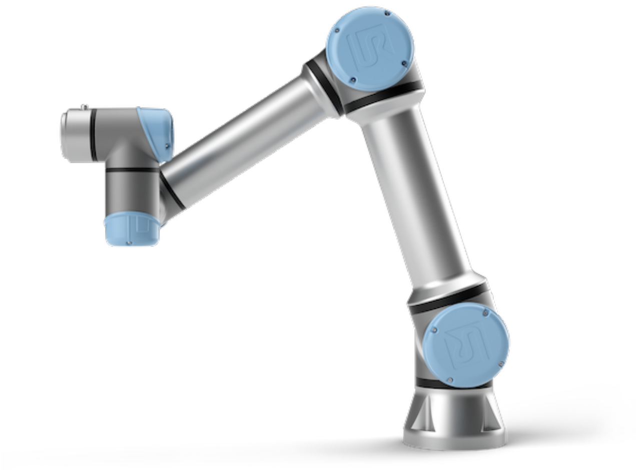


Fig. 3.5 Robot UR5 e-series.

Table 3.2 Modified DH parameters of UR5 e-serie.

Joint	$\theta_{DH}[rad]$	$a_{DH}[m]$	$d_{DH}[m]$	α_{DH} [rad]
1	θ_1	0	0.1625	$\frac{\pi}{2}$
2	θ_2	0	0	0
3	θ_3	-0.425	0	0
4	θ_4	-0.3922	0.1333	$\frac{\pi}{2}$
5	θ_5	0	0.0997	$-\frac{\pi}{2}$
6	θ_6	0	0.0996	0

3.3 Mathematical model of the mobile robot

The reference systems employed for defining the model of the AMR MiR250 are described herein. From this point forward, the MiR250 will be referred to simply as MiR. In figure 3.6, the reference systems necessary for the definition of the model of the robot are shown. In particular:

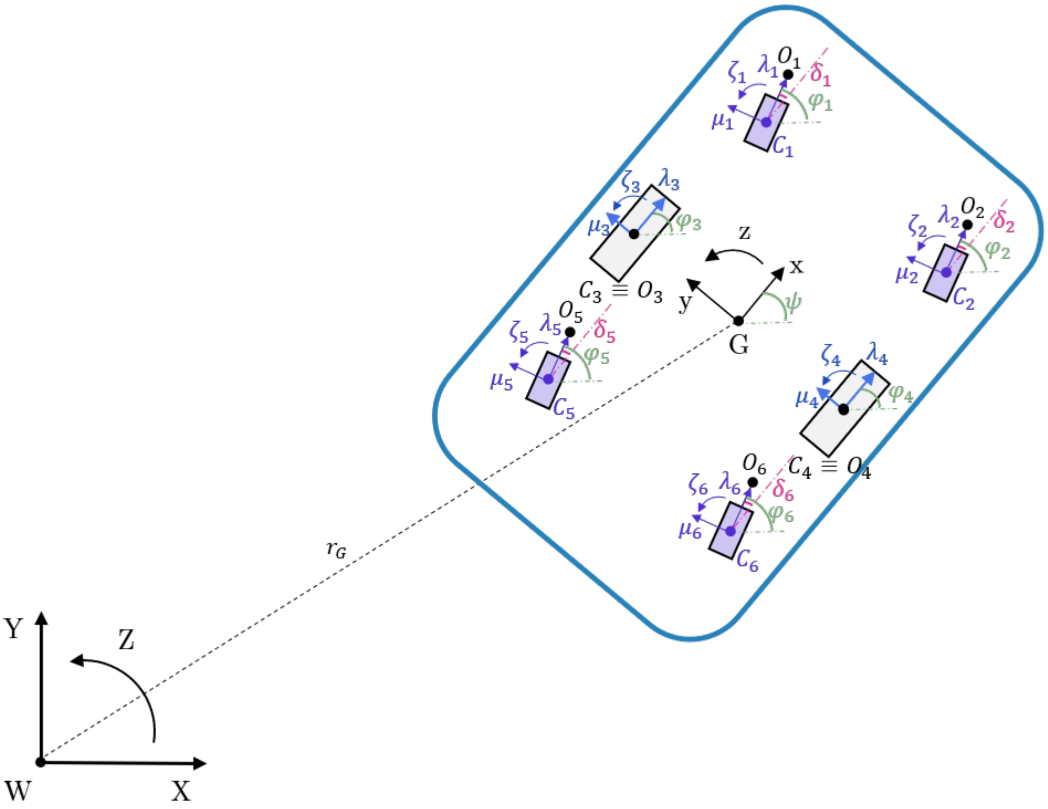


Fig. 3.6 Block diagram of MiR250 robot.

- the global reference frame, $W - XYZ$, with origin in W, which is a fixed reference frame.
- the reference systems of the robot, $G - xyz$, with origin in G, which is a mobile reference frame.
- the reference systems of the six wheels, $C_i - \lambda_i \mu_i \xi_i$, with $i = 1$ for the left front caster wheel, $i = 2$ for the right front caster wheel, $i = 3$ for the left driving wheel, $i = 4$ for the right driving wheel, $i = 5$ for the rear left caster wheel and $i = 6$ for the rear right caster wheel, which are all mobile reference frames.

Additionally, the following relative rotations can be identified:

- ψ: rotation of $G - xyz$ with respect to $W - XYZ$.
- φ_i: rotation of $C_i - \lambda_i \mu_i \xi_i$ with respect to $W - XYZ$.
- δ_i: rotation of $C_i - \lambda_i \mu_i \xi_i$ with respect to $G - xyz$.

Besides, it is important to underline that the points C_i are the points where the rotation axes of the wheels pass, while the points O_i are the points where the wheels are hinged to the frame of the MiR. O_i and C_i coincides for the driving wheels, while do not for caster wheels, so it necessary to consider the offset $\overline{C_i O_i} = b_c$.

3.3.1 Kinematics of the mobile robot

The kinematics of the mobile robot is defined by the linear velocity $\overline{V}_G$ and the angular velocity $\dot{\psi}_G$. They can be calculated in the MiR reference system $G - xyz$. Once obtained the values of the velocities, position values can be calculated, and be transformed in the global reference frame $W - XYZ$, by the rotation matrix:

$$\left\{ \begin{array}{c} X \\ Y \end{array} \right\} = \left[\begin{array}{cc} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{array} \right] \left\{ \begin{array}{c} x \\ y \end{array} \right\} \tag{3.4}$$

Deriving, the velocity formulation is obtained as:

$$\left\{ \begin{array}{c} \dot{X} \\ \dot{Y} \end{array} \right\} = \dot{\psi} \left[\begin{array}{cc} -\sin\psi & -\cos\psi \\ \cos\psi & -\sin\psi \end{array} \right] \left\{ \begin{array}{c} x \\ y \end{array} \right\} + \left[\begin{array}{cc} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{array} \right] \left\{ \begin{array}{c} \dot{x} \\ \dot{y} \end{array} \right\} \tag{3.5}$$

and deriving again, the acceleration:

$$\left\{\begin{array}{c}\ddot{X}\\ \ddot{Y}\end{array}\right\}=\ddot{\psi}\left[\begin{array}{cc}-\sin\psi & -\cos\psi\\ \cos\psi & -\sin\psi\end{array}\right]\left\{\begin{array}{c}x\\ y\end{array}\right\}-\dot{\psi}^2\left[\begin{array}{cc}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{array}\right]\left\{\begin{array}{c}x\\ y\end{array}\right\}+$$
$$+2\dot{\psi}\left[\begin{array}{cc}-\sin\psi & -\cos\psi\\ \cos\psi & -\sin\psi\end{array}\right]\left\{\begin{array}{c}\dot{x}\\ \dot{y}\end{array}\right\}+\left[\begin{array}{cc}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{array}\right]\left\{\begin{array}{c}\ddot{x}\\ \ddot{y}\end{array}\right\} \tag{3.6}$$

3.3.2 Wheel-floor Friction Model

The performance of the MiR is significantly influenced by the frictional forces that arise at the contact surface between the wheels and the plane on which movement occurs. The frictional forces between the floor and the vehicle are modeled on the principles used in tire dynamics, according to Pacejka [75]. Pacejka model, or magic formula, is a semi-empirical formulation to evaluate the friction forces acting on the tyre. These forces can act in three ways:

- Longitudinal friction force: this component generates the traction force of the vehicle.
- Lateral friction force: this component allows the vehicle to turn, as it generates the centripetal force directed towards the center of the curve.
- Self-aligning torque: this is a torque generated on the wheel that allows the axis λ_i to align with the velocity $\overline{V}_{C,i}$ of the hinge of the i_{th} wheel.

Longitudinal Friction Force

The value of the longitudinal friction force $\boldsymbol{F}_{\lambda}$ can be determined by calculating the slip ratio σ for each wheel. This slip ratio σ_i is used to compute the corresponding friction force $\boldsymbol{F}_{\lambda,i}(\sigma_i)$ acting on the wheel, as derived from experimental data.
The slip ratio is defined as:

$$\sigma_i=\frac{\dot{\theta}_i\cdot r_i-V_{Ci,\lambda i}}{\max(\dot{\theta}_i\cdot r_i,V_{Ci,\lambda i})}\qquad \sigma_i\in[-1,1] \tag{3.7}$$

where $\dot{\theta}_i \cdot r_i$ represents the peripheral speed of the wheel, i.e., the speed of the wheel at the point of contact with the surface, while $V_{Ci,\lambda i}$ denotes the speed of the wheel hub along the direction λ_i.

In addition to the slip coefficient, the longitudinal force also depends on the normal load exerted on the mobile robot.

With these premises, a simplified shape of the curve for the definition of the longitudinal forces of the wheels of the mobile robot was defined starting from Pacejkas experimental curves. It is defined by points A,B, C and D in figure 3.7 and it is composed by straight lines. The curve is identified by the degrees of freedom:

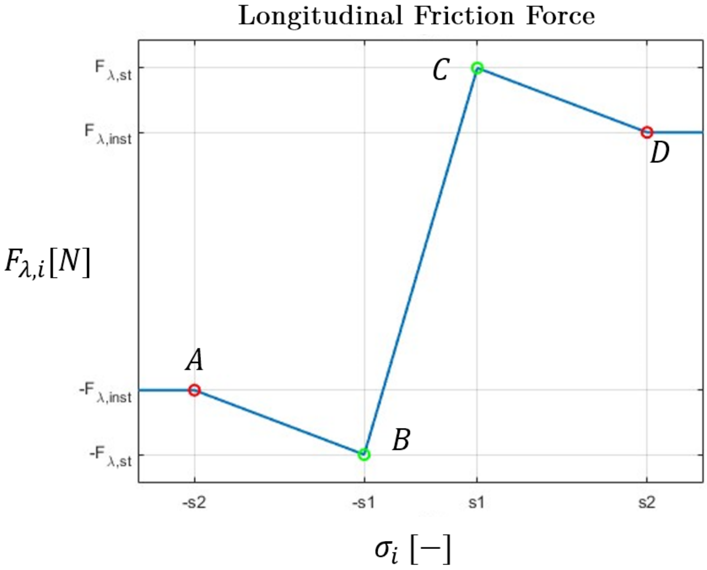


Fig. 3.7 Longitudinal Friction Force.

- Slippage s_1: defines the stability region, i.e., the interval of the domain in which the friction force increases linearly with slippage. At s_1, the longitudinal friction force reaches its maximum value, denoted as $F_{\lambda,st}$.
- Slippage s_2: from s_2 onward, the friction force remains constant and is denoted as $F_{\lambda,inst}$.
- Friction coefficient f_a: used to calculate the friction force generated under stable conditions, i.e., for $-s_1 \leq \sigma_i \leq s_1$ where the slippage is very low.
- Friction coefficient f: used to calculate the longitudinal friction force generated under unstable conditions, i.e., for $\sigma_i \leq -s_2$ and $\sigma_i \geq s_2$.

The values of $F_{\lambda,st}$ and $F_{\lambda,inst}$ depends on the normal load. As will be discussed later on, the values of these variables will be better determined with experimental tests.

Lateral Friction Force

The function that describes the lateral friction force $\boldsymbol{F}_{\mu}$ acting on the i_{th} wheel is dependent on an angle α_i, known as the slip angle. It represents, for each wheel, the inclination of the velocity vector $\overline{V}_{Ci}$ with respect to the versor λ_i. This configuration captures the deviation of the wheel motion direction from its alignment. For a better understanding, the wheel with the velocity components is shown in figure 3.8. The slip angle α_i is defined as:

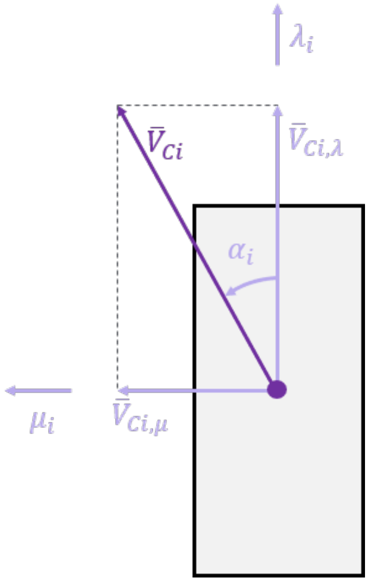


Fig. 3.8 Wheel velocity components.

$$\alpha_i = \arcsin\left(\frac{\overline{V}_{Ci,\mu}}{\overline{V}_{Ci}}\right) \tag{3.8}$$

The lateral friction force depends on the normal force, too. As for the longitudinal force, the shape of the curve for the definition of the lateral force has been obtained by simplifying Pacejkas experimental curves. A representative curve of the behaviour experimentally identified is shown in figure 3.9. Once again the curve is composed by straight lines and it is identified by point E,F,G and H in figure 3.9.The parameters to identify for curve definition are:

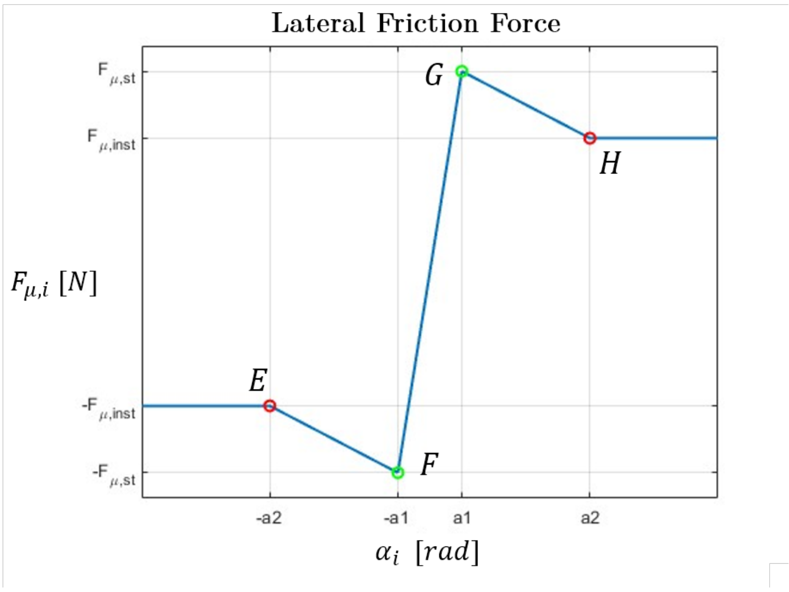


Fig. 3.9 Lateral Friction Force.

- The slip angle a_1: it defines the stability region, namely the interval within the domain where the friction force increases linearly with the slip angle. At a_1, the lateral friction force reaches its maximum value, denoted as $F_{\mu,st}$.
- The slip angle a_2: beyond a_2, the friction force remains constant and is represented as $F_{\mu,inst}$.
- The friction coefficient f_a: This coefficient is the same one used to calculate the longitudinal friction force. It is utilized to compute the tangential friction force under stable conditions, specifically within the range $-a_1 \leq \alpha_i \leq a_1$.
- The friction coefficient f: Also used in the calculation of the longitudinal friction force, it is employed to replicate the behavior of tangential friction during stick-slip conditions occurring within $-a_2 \leq \alpha_i \leq -a_1$ and $a_1 \leq \alpha_i \leq a_2$.

Self-Aligning Torque

The self-aligning torque $\boldsymbol{M}$ is generated by a shift in the line of action of the tangential friction force, which does not pass through the wheel point of contact with the surface. This shift creates a moment that tends to align the wheel with the direction of the wheel center velocity. The value of the self-aligning torque depends on the slip angle α_i, defined for the lateral friction force, and on the normal force. As previously done with the other friction contributions, the shape of the momentum is taken simplifying Pacejka experimental curves, that is shown in figure 3.10. The values that can influence the self-aligning torque curve are:

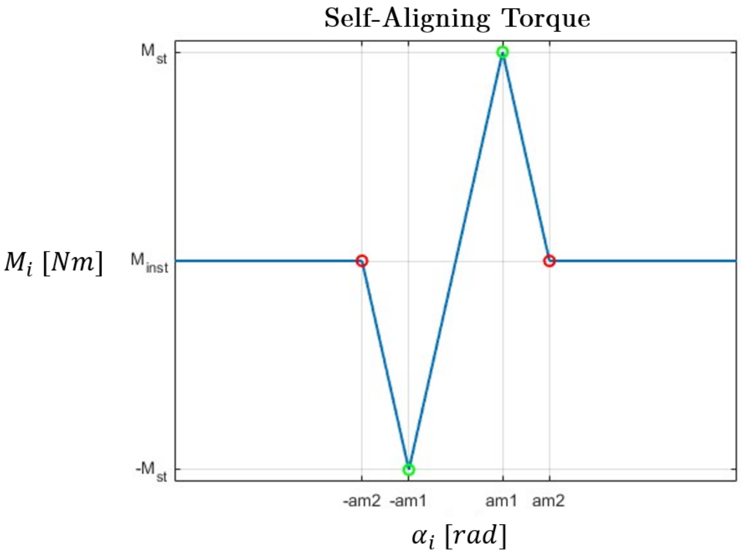


Fig. 3.10 Self-Aligning Torque.

- The slip angle $a_{m,1}$: it defines the stability region, that is, the interval within the domain where the self-aligning torque increases linearly with the slip angle. At $a_{m,1}$, the torque reaches its maximum value, denoted as M_{st}.
- The slip angle $a_{m,2}$: beyond $a_{m,2}$, the self-aligning torque remains constant and at the value M_{inst} .
- The friction coefficient f_a: the same coefficient used in the calculation of longitudinal and lateral friction forces. It is employed to compute the self-aligning torque under adherent conditions, specifically within the range $-a_{m,1} \leq \alpha_i \leq a_{m1}$.
- The friction coefficient f: used for calculating the longitudinal and lateral friction forces, it is utilized to describe the evolution of the self-aligning torque under stick-slip conditions for $-a_{m2} \leq \alpha_i \leq -a_{m1}$ and $a_{m1} \leq \alpha_i \leq a_{m2}$.

Friction Force Versus

So far, the magnitudes of the friction forces have been discussed without considering their versus. So, the versus of the friction forces is analyzed relative to the local reference system of the wheel. In figure 3.11, the friction forces of the wheel are presented. The directions of the forces are aligned with the unit vectors of the local reference system of the i_{th} wheel. In the figure, only the longitudinal friction force, the lateral friction force and the self-aligning torque are reported, in green. According to this arbitrary choice of the directions of the friction forces, the following relationships hold:

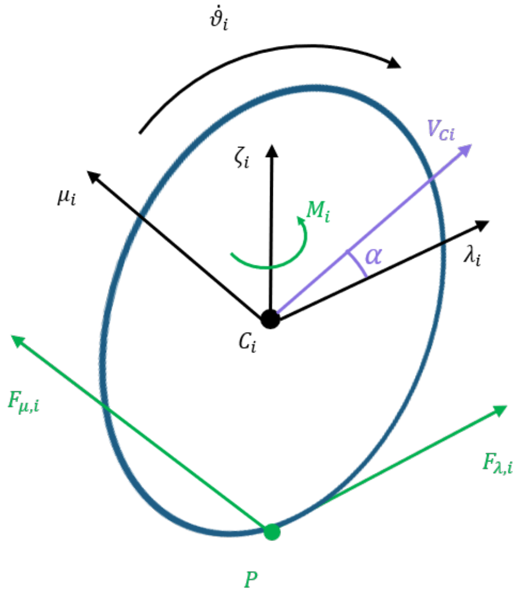


Fig. 3.11 Friction forces on a robot wheel.

$$\begin{cases} \boldsymbol{F}_{\lambda,i} = \boldsymbol{F}_{\lambda,i}(\sigma_i) \\ \boldsymbol{F}_{\mu,i} = -\boldsymbol{F}_{\mu_i}(\alpha_i) \\ \boldsymbol{M}_i = \boldsymbol{M}_i(\alpha_i) \end{cases} \tag{3.9}$$

3.3.3 Driving Wheels

The driving wheels are responsible for the motion of the mobile robot. They are interconnected by a chassis and powered by two electric motors. Unlike caster wheels, the driving wheels lack a vertical axis around which they can rotate, meaning the robot ability to turn depends on the differential in angular velocity between the wheels.

Kinematic analysis

To compute the kinematic relationships, it is necessary introduce the reference systems $O_i - \lambda_i \mu_i \zeta_i$, with $i = 3,4$. Referring to figure 3.12, the position of the driving wheel reference frames with respect to the MiR reference frame is given by the homogeneous transformation matrices:

$$O_3 - x_3^{'} y_3^{'} z_3^{'} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b_d \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x & y - b_d & 1 \end{bmatrix} \tag{3.10}$$

$$O_4 - x_4^{'} y_4^{'} z_4^{'} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b_d \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x & y + b_d & 1 \end{bmatrix} \tag{3.11}$$

Given that the system under consideration is capable of imparting a set rotational speed to the driving wheels, it is essential to derive these speeds from the velocity of the center of mass $\overline{V}_G$ and the angular velocity $\dot{\psi}$ of the MiR. Referring to figure 3.12, the velocity of the wheel hubs can be calculated using the following relation:

$$\begin{cases} V_{C3,\lambda} = V_{G,x} - \dot{\psi} \cdot b_d \\ V_{C3,\mu} = V_{G,y} \\ V_{C4,\lambda} = V_{G,x} + \dot{\psi} \cdot b_d \\ V_{C4,\mu} = V_{G,y} \end{cases} \tag{3.12}$$

in which b_d is half the chassis between the driving wheels.

Dynamic analysis

The driving wheels are actuated by electric motors, that provide the driving torques. The motors are modelled as DC motors, with the RL-circuit equations, whose circuit is shown in figure 3.13. The equations are:

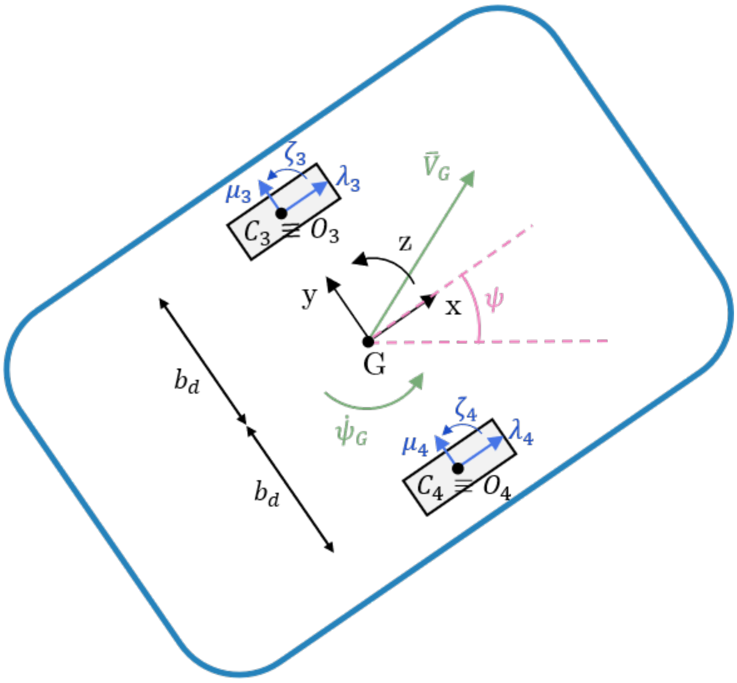


Fig. 3.12 Kinematic analysis scheme for driving wheels.

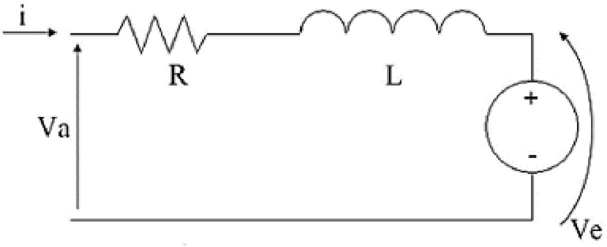


Fig. 3.13 RL-circuit of the electric motors.

$$\begin{aligned} V_a - V_e &= i \cdot R + \tfrac{di}{dt} \cdot L \\ V_e &= k_c \cdot \dot{\vartheta} \end{aligned} \tag{3.13}$$

where V_a represents the armature voltage that powers the electric motor, V_e represents the electromotive force, i is the current, k_c is the velocity constant of the motor, $\dot{\vartheta}$ the angular velocity of the motor, and R and L respectively represent the resistance and inductance of the circuit. The motor generates a torque C_m that causes the wheel to rotate. In figure 3.14 is the free body diagram (FBD) of the i_{th} driving wheel in the plane orthogonal to its axis of rotation. In the figure, the reaction forces on the hub and the normal force are not reported, for a better readability. Only the actions influencing the rotational behavior are presented. The torque generated by the electric motor is:

$$C_{m,i} = i \cdot K_t \tag{3.14}$$

where K_t is the torque constant of the motor. From the FBD, the following dynamic equilibrium can be derived:

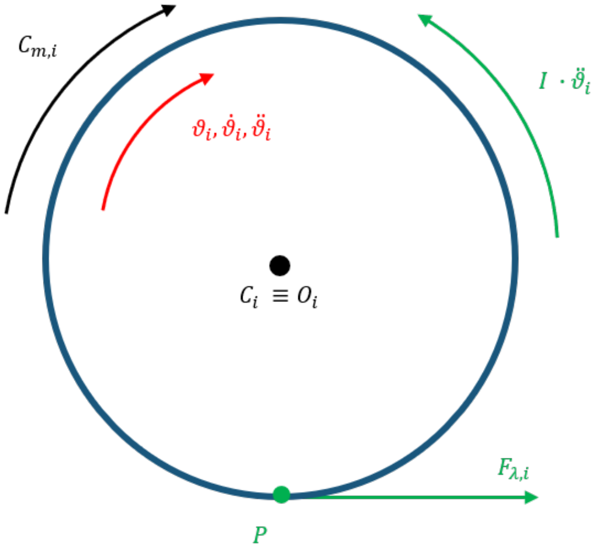

$C_{m,i}$
$\vartheta_i, \dot{\vartheta}_i, \ddot{\vartheta}_i$
$I \cdot \ddot{\vartheta}_i$
$C_i \equiv O_i$
$F_{\lambda,i}$
P

Fig. 3.14 FBD of the driving wheel.

$$C_{m,i} - I \cdot \ddot{\theta}_i - F_{i,\lambda} \cdot r_d = 0 \tag{3.15}$$

where $C_{m,i}$ is the torque produced by the motor on the $i_t h$ wheel, I is the moment of inertia of the wheel, $\ddot{\theta}_i$ is the angular acceleration of the wheel, $F_{\lambda,i}$ is the longitudinal friction force acting on the i_{th} wheel, and r_d is the radius of the wheel.

The model integrates the velocity control of the motors of the driving wheels. It comprises two nested loops. The outer one is the velocity loop, and the inner one is the current loop. The velocity loop is constituted by a PID controller that returns a current value i_{SET} proportional to the error between the set angular velocity $\dot{\vartheta}_{SET}$ and the feedback angular velocity $\dot{\vartheta_{FB}}$. The current loop is constituted by a PID controller that returns an armature voltage V_{RIF} proportional to the current error between the i_{SET} and the feedback current i_{FB}, coming from the model of the motor. The gearbox is simply modeled as a gain equal to the reduction ratio.

3.3.4 Caster Wheels

Caster wheels are used to stabilize the robot and prevent pitching movements. These wheels can freely rotate around the revolute joint that connects them to the frame at O_i. For each caster wheel, it is necessary to consider the degrees of freedom φ_i, i.e., φ_1, φ_2, φ_5 and φ_6, which depends on the rotation of the caster wheel around the vertical axis, that is δ_i. Since they are not controlled by any actuator, the dynamics of the caster wheels are governed by frictional forces.

Kinematic analysis

In order to evaluate kinematic relationships, it is necessary to introduce the reference system $O_i - x_i^{'}y_i^{'}z_i^{'}$ with $i = 1,2,5,6$, for each caster wheel. It is the reference frame around which the caster wheel turns. In this way it is possible to know the position and velocity of each caster wheel. To convert coordinates from the global reference system $G - xyz$ to the local reference system $O_i - x_i^{'}y_i^{'}z_i^{'}$ the following homogeneous transformation matrices are used:

$$O_1 - x_1^{'}y_1^{'}z_1^{'} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x-a & y-b & 1 \end{bmatrix} \tag{3.16}$$

$$O_2 - x_2^{'}y_2^{'}z_2^{'} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x-a & y+b & 1 \end{bmatrix} \tag{3.17}$$

$$O_5 - x_5^{'}y_5^{'}z_5^{'} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x+a & y-b & 1 \end{bmatrix} \tag{3.18}$$

$$O_6 - x_6^{'}y_6^{'}z_6^{'} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} x+a & y+b & 1 \end{bmatrix} \tag{3.19}$$

The data relative to caster wheels are reported in figure 3.15. For sake of simplicity, only caster wheel 1 and caster wheel 2 are drawn. As can be seen, the angular position of any rigid body relative to the local reference system $O_i - x_i^{'}y_i^{'}z_i^{'}$ is identical to that relative to the global reference system $G - xyz$. The aim is to calculate te velocity $\overline{V}_{Ci}$ in order to calculate the slip angle and the slip ratio. It results:

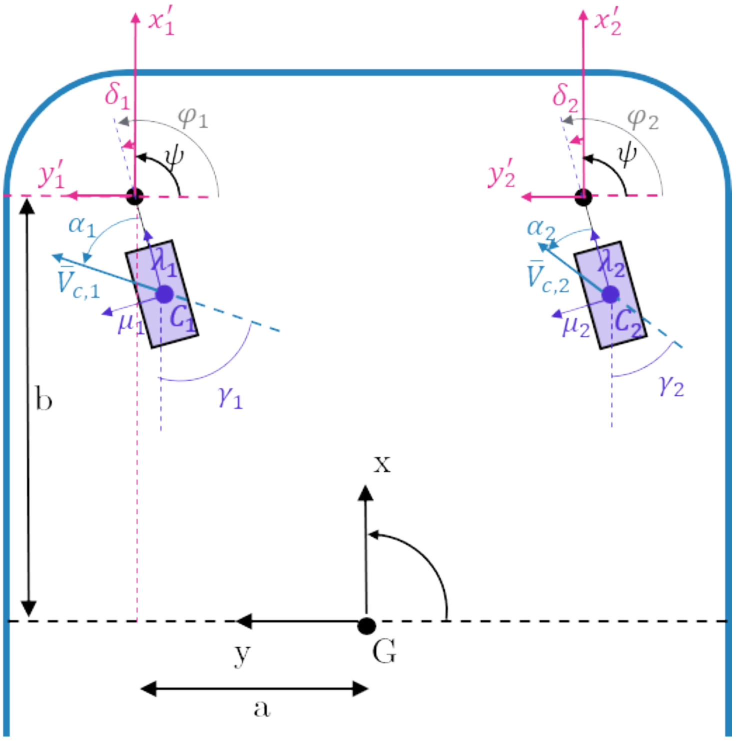


Fig. 3.15 Kinematic scheme of the front caster wheels

$$\overline{V}_{Ci} = \overline{V}_{Ci,t} + \overline{V}_{Ci,r} \tag{3.20}$$

$\overline{V}_{Ci}$ is the sum of a drag velocity $\overline{V}_{Ci,t}$ which represents the velocity that point C_i would have if it were fixed to O_i and a relative velocity $\overline{V}_{Ci,r}$ that represents the velocity of point C_i relative to the local reference frame $O_i - x_i^{'}y_i^{'}z_i^{'}$. These components are:

$$\overline{V}_{Ci,t} = \overline{V}_{Oi} + \overline{\psi} \times \overline{(C_i - O_i)} = \overline{V}_G + \overline{\psi} \times \overline{(O_i - G)} + \overline{\psi} \times \overline{(C_i - O_i)} \tag{3.21}$$

$$\overline{V}_{Ci,r} = \dot{\overline{\delta}}_i \times \overline{(C_i - O_i)} \tag{3.22}$$

and so it becomes:

$$\overline{V}_{Ci} = \overline{V}_G + \overline{\psi} \times \overline{(O_i - G)} + (\overline{\psi} + \dot{\overline{\delta_i}}) \times \overline{(C_i - O_i)} \tag{3.23}$$

and this is valid for all the caster wheels. Each vectorial equation can be scomposed, and it results:

$$\begin{cases} \overline{V}_{C1,x} = \dot{x}_G - \dot{\psi} b + (\delta_1 + \dot{\psi}) b_c \sin \delta_1 \\ \overline{V}_{C1,y} = \dot{y}_G + \dot{\psi} a - (\delta_1 + \dot{\psi}) b_c \cos \delta_1 \end{cases} \tag{3.24}$$

$$\begin{cases} \overline{V}_{C2,x} = \dot{x}_G + \dot{\psi} b + (\delta_2 + \dot{\psi}) b_c \sin \delta_2 \\ \overline{V}_{C2,y} = \dot{y}_G + \dot{\psi} a - (\delta_2 + \dot{\psi}) b_c \cos \delta_2 \end{cases} \tag{3.25}$$

$$\begin{cases} \overline{V}_{C5,x} = \dot{x}_G - \dot{\psi} b + (\delta_5 + \dot{\psi}) b_c \sin \delta_5 \\ \overline{V}_{C5,y} = \dot{y}_G - \dot{\psi} a - (\delta_5 + \dot{\psi}) b_c \cos \delta_5 \end{cases} \tag{3.26}$$

$$\begin{cases} \overline{V}_{C6,x} = \dot{x}_G + \dot{\psi} b + (\delta_6 + \dot{\psi}) b_c \sin \delta_6 \\ \overline{V}_{C6,y} = \dot{y}_G - \dot{\psi} a - (\delta_6 + \dot{\psi}) b_c \cos \delta_6 \end{cases} \tag{3.27}$$

Known the velocity components of the wheel centers, it is possible to calculate both the magnitude and the orientation relative to the $x^{'}$ axis of the reference system $O_i - x_i^{'} y_i^{'} z_i^{'}$. In figure 3.15, the orientation is denoted as γ_i:

$$\begin{cases} \gamma_i = \arctan\left(\frac{V_{Ci,y}}{V_{Ci,y}}\right) \\ V_{Ci} = \sqrt{V_{Ci,x}^2 + V_{Ci,y}^2} \end{cases} \tag{3.28}$$

Now, known the inclination γ_i of the velocity, V_{Ci} relative to the axis $x_i^{'}$, and known the inclination δ_i of the wheels relative to the axis $x_i^{'}$, one can calculate the inclination α_i of the velocity V_{Ci}, relative to the direction γ_i and calculate the component of V_{Ci} along the direction γ_i:

$$\begin{cases} \alpha_i = \gamma_i - \delta_i \\ V_{Ci,\gamma_i} = V_{Ci} \cos \alpha_i \end{cases} \tag{3.29}$$

With this information, the kinematic quantities necessary for calculating the longitudinal and lateral friction forces, and the self-aligning torques, are obtained.

Dynamic analysis

Known α_i and V_{Ci,γ_i}, it is possible to calculate the friction forces $\boldsymbol{F}_{\lambda,i}(\sigma_i)$ and $\boldsymbol{F}_{\mu,i}(\alpha_i)$ acting on the caster wheels. In this instance, the contribution of the self-aligning torque has not been considered, as it provides a negligible effect compared to the impact caused by the tangential friction force $\boldsymbol{F}_{\mu,i}(\alpha_i)$. In figure 3.16, the free body diagram of a caster wheel is shown. Observing figure 3.16, it is possible to calculate δ_i, which is necessary for evaluating α_i, eq. 3.29. The parameter φ_i and its derivatives refer to the angular position, velocity, and acceleration of a generic caster wheel relative to the X axis of the absolute reference system $W - XYZ$. By subtracting ψ, that is the angular position of the MiR in the absolute reference frame, the angular position of the caster wheel relative to the local reference system $G - xyz$ is obtained, that is the angle δ_i. In this way it is possible to calculate α_i, which represents the angle between the velocity direction of the center of the i_{th} wheel $\overline{V}_{C_i}$ and the λ_i axis of the reference system relative to the i_{th} wheel $C_i - \lambda_i\mu_i\varphi_i$.

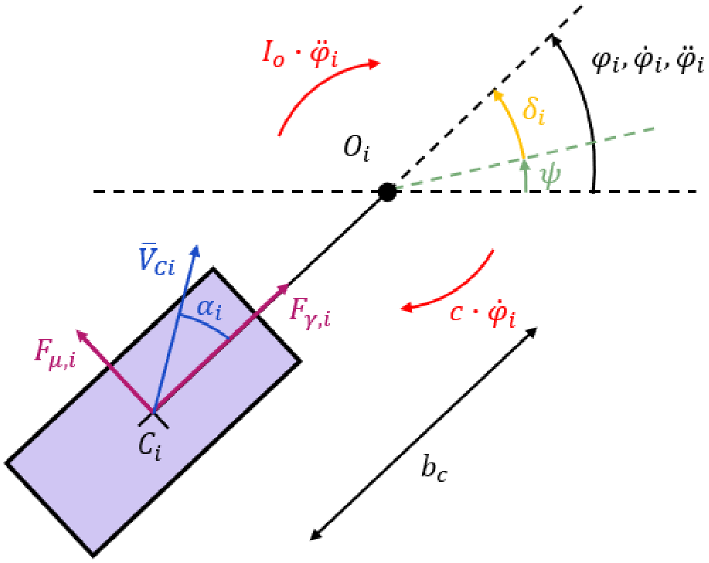

$I_o \cdot \ddot{\varphi}_i$
$\varphi_i, \dot{\varphi}_i, \ddot{\varphi}_i$
δ_i
O_i
ψ
$\bar{V}_{Ci}$
α_i
$F_{\gamma,i}$
$c \cdot \dot{\varphi}_i$
$F_{\mu,i}$
C_i
b_c

Fig. 3.16 Free body diagram of a caster wheel.

So, referring to figure 3.16, it is possible to write the momentum equilibrium equation:

$$I_{O,i} \cdot \ddot{\varphi}_i + c \cdot \dot{\varphi}_i + F_{\mu,i} b_c = 0 \tag{3.30}$$

which, is valid for all the caster wheels. The dissipative contribution is entirely concentrated in c, which represents the coefficient of viscous friction. The forces equilibrium equations are trivial and not reported.

3.3.5 Normal reaction forces

Each wheel is subjected to a normal force that determines the values of the friction forces. To calculate the normal reactions of the wheels, the FBD of the MiR is represented in the xz and yz planes of the reference system $G-xyz$.

Equilibrium equations in xz plane

In figure 3.17 the FBD of the MiR in the xz plane is shown. For simplicity, in the FBD, $\delta_i = 0$, for $i = 1,2,5,6$. In deriving the equilibrium equation, the relative rotation δ_i that each caster wheel can undergo is taken into account. For simplicity, the longitudinal friction forces $\boldsymbol{F}_{i,\lambda_i}$ are not represented, as they do not generate

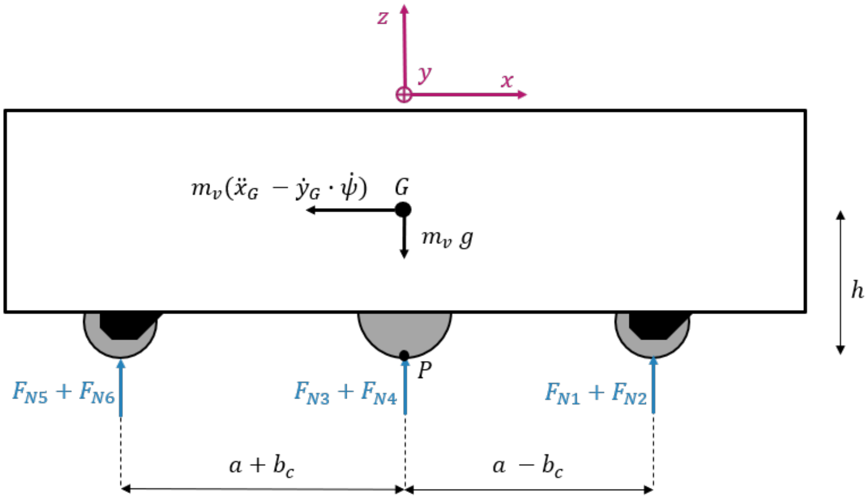


Fig. 3.17 Free body diagram of the MiR in the xz plane.

moments in the equilibrium around the pole P. The equilibrium equation is as follows:

$$\begin{aligned} m_v \cdot (\ddot{x}_G - \dot{y}_G \cdot \dot{\psi}) \cdot h = -F_{N1} \cdot (a - b_c \cdot \cos\delta_1) - F_{N2} \cdot (a - b_c \cdot \cos\delta_2) + \\ + F_{N5} \cdot (a + b_c \cdot \cos\delta_5) + F_{N6} \cdot (a + b_c \cdot \cos\delta_6) \end{aligned} \tag{3.31}$$

where m_v is the mass of the robot, $\ddot{x}_G$ is the acceleration of the center of mass in the x direction, $\dot{y}_G$ is the velocity of the center of mass in the y direction, $\dot{\psi}$ is the angular velocity, h is the height of the center of mass, $F_{i,N}$ is the normal reaction force for each wheel, and δ_i is the relative rotation angle for each caster wheel.

Equilibrium equations in xy plane

In figure 3.18 the FBD of the MiR in the xy plane is shown. Also in this case, $\delta_i = 0$, for $i = 1, 2, 5, 6$. The equilibrium equation around the pole P is:

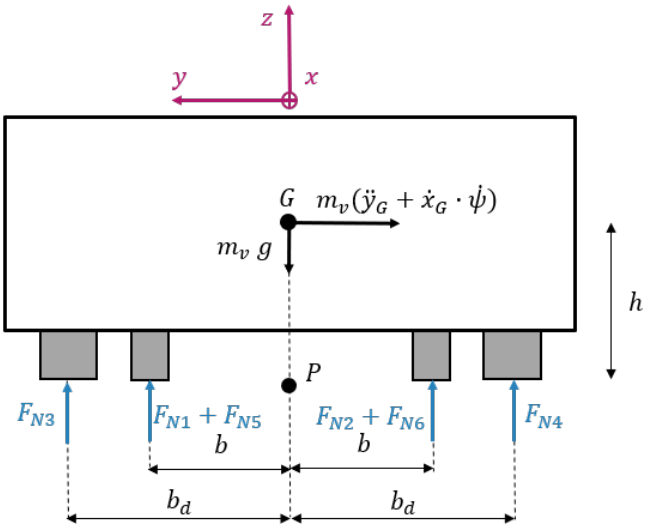


Fig. 3.18 Free body diagram of the MiR in the xy plane.

$$\begin{aligned} m_v \cdot (\ddot{y}_G + \dot{x}_G \cdot \dot{\psi}) \cdot h = -F_{N1} \cdot (b - b_c \cdot \sin\delta_1) - F_{N5} \cdot (b - b_c \cdot \sin\delta_5) - F_{N3} \cdot b_d + \\ + F_{N2} \cdot (b + b_c \cdot \sin\delta_2) + F_{N6} \cdot (b + b_c \cdot \sin\delta_6) + F_{N4} \cdot b_d \end{aligned} \tag{3.32}$$

where $\ddot{y}_G$ is the acceleration of the center of mass in the y direction and $\dot{x}_G$ is the velocity of the center of mass in the x direction. Also the equilibrium equation along the z axis is derived as:

$$m_v \cdot g = F_{N1} + F_{N2} + F_{N3} + F_{N4} + F_{N5} + F_{N6} \tag{3.33}$$

Three more equations are necessary to close the system, and they are derived by making assumptions. Starting from figure 3.18, the MiR can be considered as a double portal frame with an equally distributed load. The inner pillars (which in the MiR are analogous to the driving wheels) apply a greater reaction force compared to the pillars at the ends. This leads to the following assumption:

$$F_{N3} + F_{N4} = F_{N1} + F_{N2} + F_{N5} + F_{N6} \tag{3.34}$$

The last two assumptions regards the caster wheels. It is assumed that the reactions of the front wheels are equal to each other, as are the reactions of the rear wheels, namely:

$$F_{N1} = F_{N2} \quad \text{and} \quad F_{N5} = F_{N6} \tag{3.35}$$

3.3.6 Dynamic of the complete model

In figure 3.19 the complete free body diagram of the MiR is shown. As can be seen, instead of considering the friction forces acting in the longitudinal direction $F_{\lambda,i}$ and in the tangential direction $F_{\mu,i}$ on the wheels, their projections in the x and y directions of the relative reference system $G-xyz$ are considered, i.e., $F_{x,i}$ and $F_{y,i}$ respectively. In the figure, the self-aligning torques M_3 and M_4 on the driving wheels, the inertial force divided in the contributions along x and y directions, i.e., $F_{x,in}$ and $F_{y,in}$, and the torque $M_{in} = I_V \cdot \ddot{\psi}$. The transformation is achieved through the rotation matrix for the generic i_{th} wheel:

$$\begin{bmatrix} F_{x,i} \\ F_{y,i} \end{bmatrix} = \begin{bmatrix} \cos\delta_i & -\sin\delta_i \\ \sin\delta_i & \cos\delta_i \end{bmatrix} \cdot \begin{bmatrix} F_{\lambda,i} \\ F_{\mu,i} \end{bmatrix} \tag{3.36}$$

In the specific case of the driving wheels, where $\delta_3 = \delta_4 = 0$, it is:

$$\begin{bmatrix} F_{x,3} \\ F_{y,3} \end{bmatrix} = \begin{bmatrix} \cos 0 & -\sin 0 \\ \sin 0 & \cos 0 \end{bmatrix} \cdot \begin{bmatrix} F_{\lambda,3} \\ F_{\mu,3} \end{bmatrix} = \begin{bmatrix} F_{\lambda,3} \\ F_{\mu,3} \end{bmatrix} \tag{3.37}$$

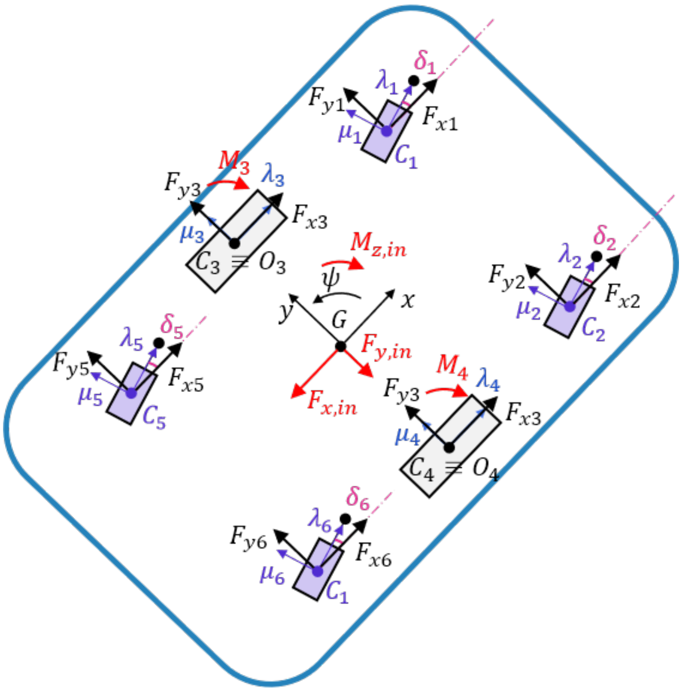


Fig. 3.19 Free body diagram of the MiR in top view.

$$\begin{bmatrix}F_{x,4}\\F_{y,4}\end{bmatrix} = \begin{bmatrix}\cos 0 & -\sin 0\\ \sin 0 & \cos 0\end{bmatrix} \cdot \begin{bmatrix}F_{\lambda,4}\\F_{\mu,4}\end{bmatrix} = \begin{bmatrix}F_{\lambda,4}\\F_{\mu,4}\end{bmatrix} \tag{3.38}$$

with δ_i the angle of the caster wheel with respect to the MiR reference frame. At this point, Newton-Euler equations are applied to derive the cardinal equations of dynamics:

$$x: F_{x,1} + F_{x,2} + F_{x,3} + F_{x,4} + F_{x,5} + F_{x,6} = F_{x,in} = m_v \cdot (\ddot{x}_G - \dot{y}_G \cdot \dot{\psi}) \tag{3.39}$$

$$y: F_{y,1} + F_{y,2} + F_{y,3} + F_{y,4} + F_{y,5} + F_{y,6} = F_{y,in} = m_v \cdot (\ddot{y}_G + \dot{x}_G \cdot \dot{\psi}) \tag{3.40}$$

$$\begin{aligned} z: &- F_{x,1}(b - b_c \sin\delta_1) + F_{x,1}(a - b_c \cos\delta_1) + F_{x,2}(b - b_c \sin\delta_2) \\ &+ F_{y,2}(a - b_c \cos\delta_2) - F_{x,3} b_d + F_{x,4} b_d + M_3 + M_4 \\ &- F_{x,5}(b - b_c \sin\delta_5) - F_{y,5}(a + b_c \cos\delta_5) + F_{x,6}(b + b_c \sin\delta_6) \\ &- F_{y,6}(a + b_c \cos\delta_6) = M_{z,in} = I_V \cdot \ddot{\psi} \end{aligned} \tag{3.41}$$

Once the laws that describe the system dynamics are known, one can explicitly define $\ddot{x}_G$, $\ddot{y}_G$, and $\ddot{\psi}$, implement the equations in a numerical solver (e.g., in Simulink), and derive the time evolution of the three coordinates $x_G(t)$, $y_G(t)$, and $\psi(t)$.

3.4 Mathematical model of the antropomorphic arm

The dynamic model of the collaborative UR5 e-series robot was developed in a multibody environment. The simulation environment enables the modeling of mechanical systems using blocks that represent bodies, joints, constraints, force elements, and sensors. The simulation tool formulates and solves the equations of motion for the complete mechanical system. In figure 3.20, the dynamic model developed is shown. The simulation environment allows for the importation of CAD geometries, including masses, inertias, joints, constraints, and 3D geometries. An automatically generated 3D animation facilitates the visualization of the system dynamics. It also enables the development of control systems and the testing of system-level performance.

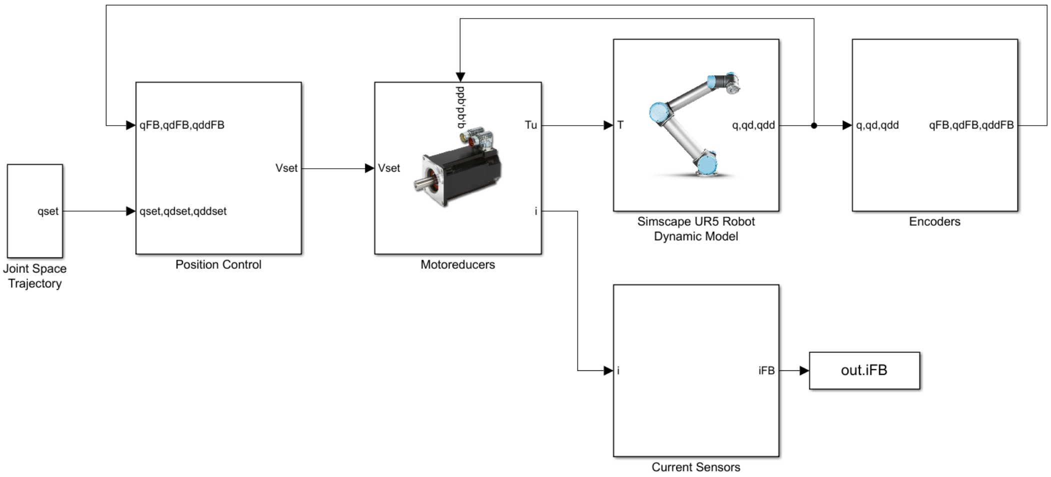


Fig. 3.20 UR5 dynamic model in multibody environment.

The model represents the significant components of the UR5 e-series robot, which we will refer to simply as UR5 for convenience, integrating a position control system. The position control operates in the joint space. Four macro-systems can be identified:

- Set trajectory and position control.
- Gearmotors.
- Joint dynamics.
- Sensors, i.e., encoders and current sensors.

The Joint Space Trajectory subsystem specifies the set angular positions q_{set}, velocities $\dot{q}_{set}$, and accelerations $\ddot{q}_{set}$ for all six joints of the robot. These parameters serve as inputs to the Position Control block, where the control mechanisms for each joint are delineated. In the Position Control system, the set position value of each joint q_{set} is subtracted from the feedback position value q_{FB} calculated within the Dynamic Model Block of the joints. The resulting error undergoes processing by a Proportional-Integral (PI) control algorithm and is further multiplied by the motor driver gain. This process yields the motor voltage value V_{set}, which serves as an input to the Gearmotor subsystem.

Within the Gearmotor block, the set voltage V_{set} and the angular velocity of the robot joints $\dot{q}$ enter. This block is divided into models of the motors and the gearboxes for all joints. The motors are modeled as DC motors based on RL-circuit equations, with specific data derived from the motor datasheets. Conversely, the gearboxes are modeled to represent the reduction ratio and the efficiency of each respective gearbox through a gain factor. Ultimately, this block outputs the torque values for each of the six joints, integrating the critical aspects of motor and gearbox dynamics in the robot overall motion control strategy. The dynamics of the robot joints is modeled using the multibody software blocks. The dynamic parameters are those identified by Raviola.

3.5 Complete model identification and validation

Once the models of the two robots have been presented, it is possible to integrate them to obtain the model of the robotic manipulator. Subsequently, the experimental setup used will be described in order to proceed with the identification and subsequent validation of the robotic system.

3.5.1 Models integration

The complete model preserves the structure of the model of the mobile robot and adds the contributions of the cabinet, that is placed on the MiR base, and the contribution of the UR5. The interaction translates into an exchange of forces through the constraint that connects the base of the UR5 to the MiR. Between the MiR and the UR5, there is an intermediary cabinet, with a height H from the floor, which contains the control system of the UR5. Figure 3.21 shows the FBD of the MiR with the control cabinet, where all the forces from the UR5 act at point T. Given that the UR5 is connected to the MiR through a fixed joint, which constrains three rotations and three translations between two rigid bodies in space, there are three forces, R_x, R_y, and R_z, and three torques, M_{Rx}, M_{Ry}, and M_{Rz}, acting on the MiR. These forces and torques are equal and opposite to those acting at the base of the UR5 to prevent relative motion between link 0, which belongs to the UR5, and the MiR. The forces and torques are directly derived from the model of the UR5. Referring to 3.22, it is possible to calculate the dynamic equilibrium of the MiR:

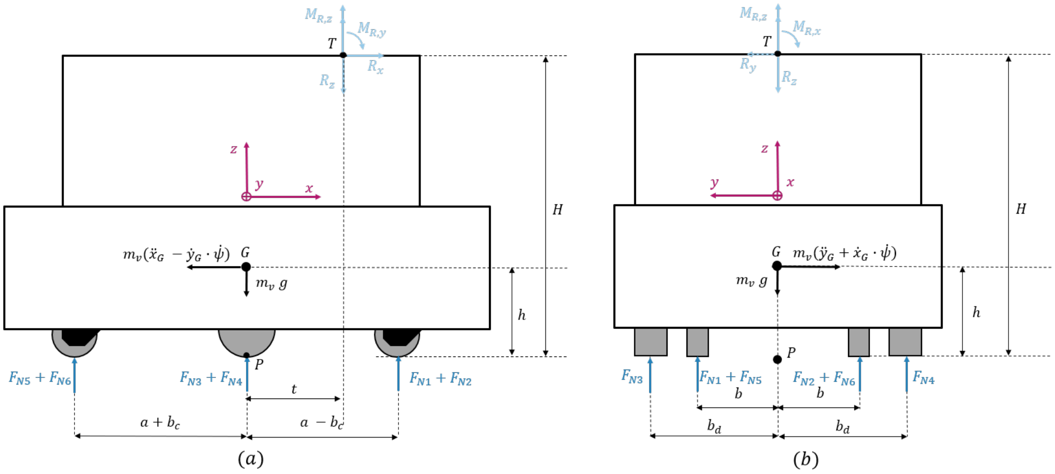


Fig. 3.21 FBD of the complete system, (a) lateral view and (b) rear view.

$$x : F_{x,1} + F_{x,2} + F_{x,3} + F_{x,4} + F_{x,5} + F_{x,6} + R_x = F_{x,in} = m_v \cdot (\ddot{x}_G - \dot{y}_G \cdot \dot{\psi}) \tag{3.42}$$

$$y : F_{y,1} + F_{y,2} + F_{y,3} + F_{y,4} + F_{y,5} + F_{y,6} + R_y = F_{y,in} = m_v \cdot (\ddot{y}_G + \dot{x}_G \cdot \dot{\psi}) \tag{3.43}$$

$$\begin{aligned} z :& M_{Rz} + R_y \cdot t - F_{x,1}(b - b_c \sin\delta_1) + F_{x,1}(a - b_c \cos\delta_1) + F_{x,2}(b - b_c \sin\delta_2) \\ &+ F_{y,2}(a - b_c \cos\delta_2) - F_{x,3} b_d + F_{x,4} b_d + M_3 + M_4 \\ &- F_{x,5}(b - b_c \sin\delta_5) - F_{y,5}(a + b_c \cos\delta_5) + F_{x,6}(b + b_c \sin\delta_6) \\ &- F_{y,6}(a + b_c \cos\delta_6) = M_{z,in} = I_V \cdot \ddot{\psi} \end{aligned} \tag{3.44}$$

in which F_x, i and F_y, i are the longitudinal and tangential friction forces $F_{\lambda,i}$ and $F_{\mu,i}$ along the $x - axis$ and $y - axis$, respectively, according to 3.36, 3.37 and 3.38, M_i are the self-aligning torques on the driving wheels, $F_{x,in}$ and $F_{y,in}$ are the inertial force divided in the contributions along x and y directions, and the torque $M_{in} = I_V \cdot \ddot{\psi}$. Referring, instead to figure 3.21, it is possible to calculate the normal reactions acting

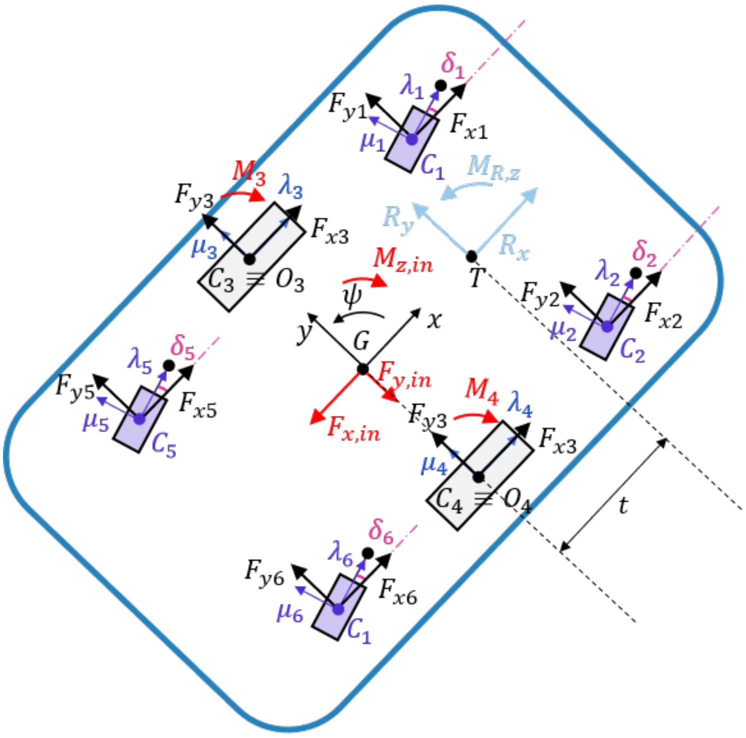


Fig. 3.22 FBD of the complete system, top view.

on the wheels, considering only the disturbances of the UR5 forces. These values are added to the values found in section 3.3.5, according to superposition principle. Referring to figure 3.21(a), it is possible to calculate:

$$\begin{aligned} y: & F_{N,1} \cdot (a - b_c \cdot \cos\delta_1) + F_{N,2} \cdot (a - b_c \cdot \cos\delta_2) + \\ & - F_{N,5} \cdot (a + b_c \cdot \cos\delta_5) - F_{N,6} \cdot (a + b_c \cdot \cos\delta_6) + \\ & - R_z \cdot t - M_{Ry} = 0 \end{aligned} \tag{3.45}$$

$$z: \; R_z = F_{N,1} + F_{N,2} + F_{N,3} + F_{N,4} + F_{N,5} + F_{N,6} \tag{3.46}$$

while, referring to figure 3.21(b):

$$\begin{aligned} x: & R_y \cdot H - M_{Rx} - F_{N,1} \cdot (b - b_c \cdot \sin\delta_1) - F_{5,N} \cdot (b - b_c \cdot \sin\delta_5) - F_{N,3} \cdot b_d + \\ & + F_{2,N} \cdot (b + b_c \cdot \sin\delta_2) + F_{N,6} \cdot (b + b_c \cdot \sin\delta_6) + F_{N,4} \cdot b_d = 0 \end{aligned} \tag{3.47}$$

In addition, the assumptions already made are still valid, so:

$$F_{N,3} + F_{N,4} = F_{N,1} + F_{N,2} + F_{N,5} + F_{N,6} \tag{3.48}$$

$$F_{N,1} = F_{N,2} \tag{3.49}$$

$$F_{N,5} = F_{N,6} \tag{3.50}$$

Once solved the equations, the values found are added to the normal forces found according to 3.3.5.

3.5.2 Experimental setup

To identify and validate the model, tuning of parameters was performed based on experimentally acquired data. The laboratory setup is shown in figure 3.23, where the mobile robot MiR250, with the collaborative robot UR5 mounted on it, can be seen. An aluminum box housing the UR5 electronics, including the controller, input/output interfaces, and a wireless hub for remote communication, is interposed between the two. The area is observed by an Optitrack vision system equipped

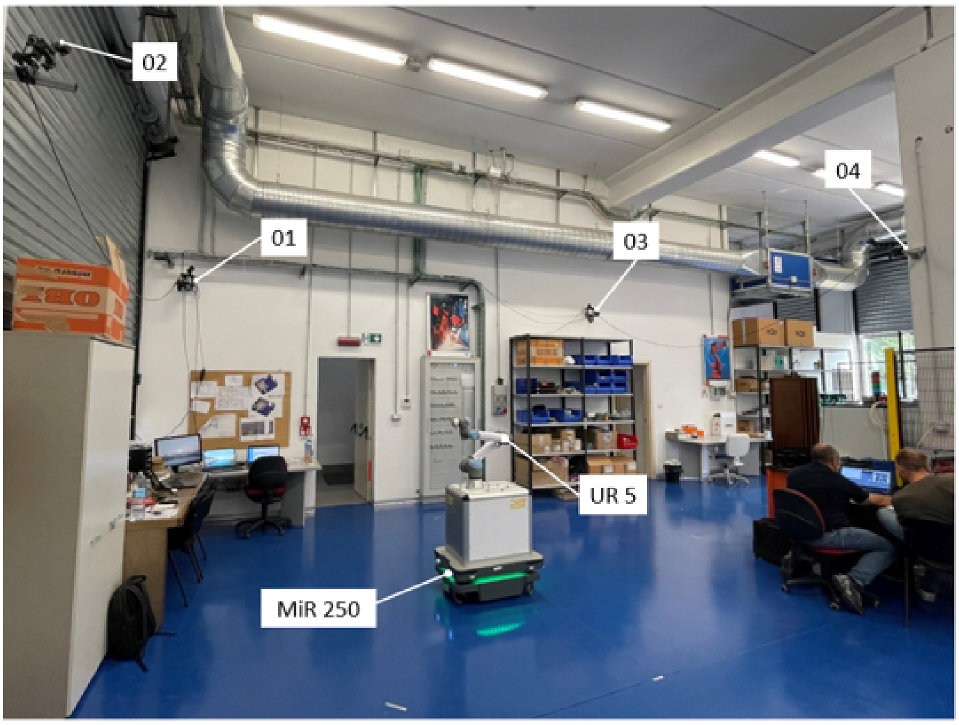
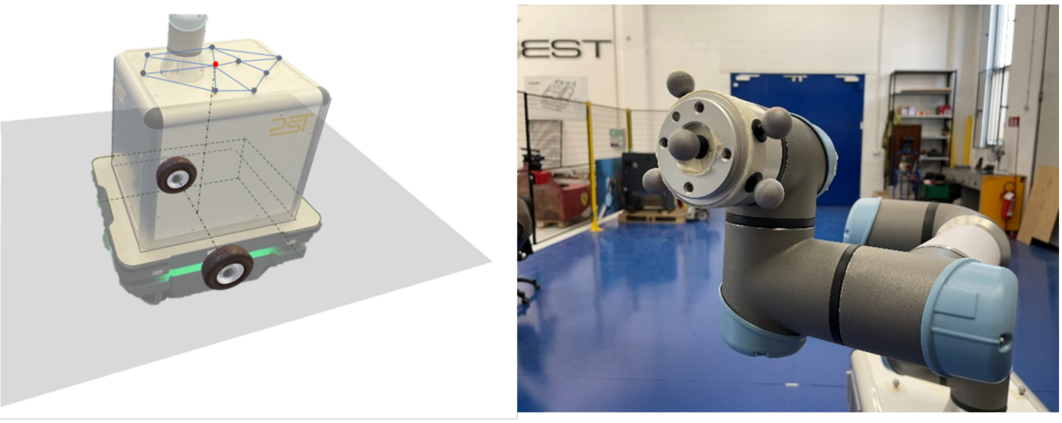


Fig. 3.23 Laboratory layout.

with 4 cameras, identified as $O1$, $O2$, $O3$, and $O4$. The vision system is based on the tracking of passive markers and can measure their position with an error of 0.5 *mm* and a frequency of 120 fps. By appropriately placing at least three markers on each part of interest, it is possible to measure the position and orientation of the body. the markers have been applied to track the pose of the MiR and the end-effector of the robotic arm, as shown in figure 3.24, (a) for the MiR robot and (b) for the UR5 end effector. The layout is schematized in figure 3.25. The Optitrack measurements are provided in the $O_p - x_p y_p z_p$ reference system, obtained during the calibration procedure of the 4 cameras. The feedback data of the mobile robot are instead retrieved with respect to the MiR reference frame $O_M - x_M y_M z_M$. In fact, usually, these two frames do not coincide. The comparison of the data coming from the various sensors is possible following the spatial calibration procedure, which consists of finding the matrix that allows transforming the data acquired by MiR in the Optitrack reference system. Optitrack markers are placed on the upper face of the box on which the UR5 base is mounted. Markers are arranged to create an asymmetrical geometry that is easily recognized by Optitrack. These markers define a rigid body to which the MiR pose is associated. In particular, the position of the MiR is defined by the marker which, considering the projection of the robot onto the support surface, identifies the center line of the drive wheels, figure 3.24(a).



(a) (b)

Fig. 3.24 Marker positions for tracking (a) the MiR robot and (b) the UR5 end effector.

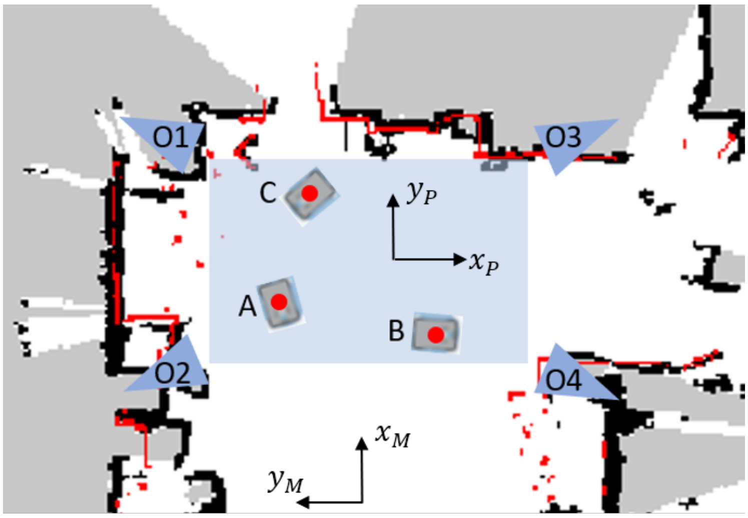


Fig. 3.25 Scheme of the experimental layout.

Spatial Calibration

The problem of spatial calibration involves finding the homogeneous matrix ${}^{p}\boldsymbol{A}_M$ that allows for the transformation of data acquired by the MiR into the Optitrack reference system. For this purpose, it is sufficient that the positions of three points are known in both the $O_p - x_p y_p z_p$ and $O_M - x_M y_M z_M$ reference systems. This allows the definition of an auxiliary frame $O_a - x_a y_a z_a$ that acts as a bridge between the two reference systems being calibrated, as schematized in figure 3.26. The matrix ${}^{p}\boldsymbol{A}_M$

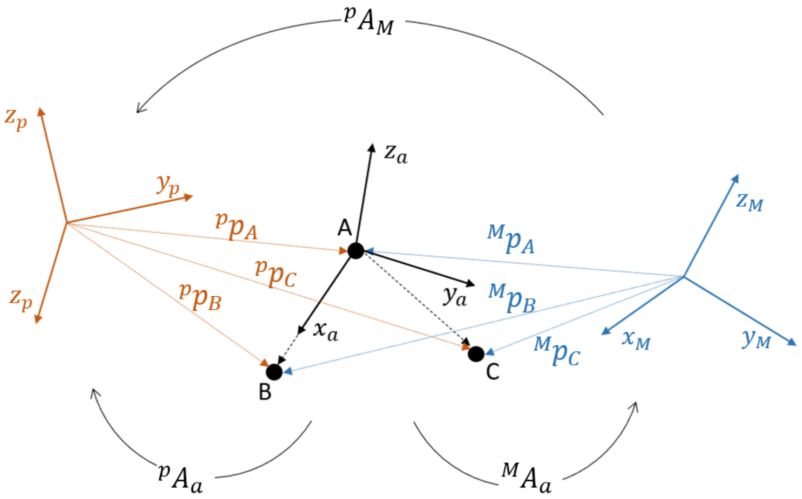


Fig. 3.26 Scheme of the experimental layout.

is calculated as:

$$ {}^{p}\boldsymbol{A}_{M} = {}^{p}\boldsymbol{A}_{a} \cdot \left({}^{M}\boldsymbol{A}_{a}\right)^{-1} \tag{3.51} $$

where ${}^{p}\boldsymbol{A}_{a}$ is the transformation matrix from the auxiliary reference frame to the Optitrack reference frame, and ${}^{M}\boldsymbol{A}_{a}$ denotes the transformation matrix from the auxiliary reference frame to the MiR reference frame. The calculation of ${}^{p}\boldsymbol{A}_{a}$ and ${}^{M}\boldsymbol{A}_{a}$ proceeds as follows. Let ${}^{p}\boldsymbol{p}_{A}$, ${}^{p}\boldsymbol{p}_{B}$, and ${}^{p}\boldsymbol{p}_{C}$ be the positions of the three points A, B, and C measured in the Optitrack reference frame, expressed as 3×1 vectors. Similarly, let ${}^{M}\boldsymbol{p}_{A}$, ${}^{M}\boldsymbol{p}_{B}$, and ${}^{M}\boldsymbol{p}_{C}$ be the positions of the same points measured in the MiR reference frame. The matrix ${}^{o}\boldsymbol{A}_{a}$ is calculated as:

$$ {}^{p}\boldsymbol{A}_{a} = \begin{bmatrix} {}^{p}\hat{\boldsymbol{x}}_{a} & {}^{p}\hat{\boldsymbol{y}}_{a} & {}^{p}\hat{\boldsymbol{z}}_{a} & {}^{p}\hat{\boldsymbol{p}}_{A} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.52} $$

where ${}^{p}\hat{\boldsymbol{x}}_{a}$, ${}^{p}\hat{\boldsymbol{y}}_{a}$ and ${}^{p}\hat{\boldsymbol{z}}_{a}$ are represent the unit vectors of the auxiliary frame expressed in the Optitrack reference system, obtained from the relations:

$$ {}^{p}\hat{\boldsymbol{x}}_{a} = \frac{{}^{p}\boldsymbol{p}_{B} - {}^{p}\boldsymbol{p}_{A}}{\|{}^{p}\boldsymbol{p}_{B} - {}^{p}\boldsymbol{p}_{A}\|} \tag{3.53} $$

$$ {}^{p}\hat{\boldsymbol{z}}_{a} = \frac{({}^{p}\boldsymbol{p}_{B} - {}^{p}\boldsymbol{p}_{A}) \times ({}^{p}\boldsymbol{p}_{C} - {}^{p}\boldsymbol{p}_{A})}{\|\,({}^{p}\boldsymbol{p}_{B} - {}^{p}\boldsymbol{p}_{A}) \times ({}^{p}\boldsymbol{p}_{C} - {}^{p}\boldsymbol{p}_{A})\,\|} \tag{3.54} $$

$$ {}^{p}\hat{\boldsymbol{z}}_{a} = {}^{p}\hat{\boldsymbol{z}}_{a} \times {}^{p}\hat{\boldsymbol{x}}_{a} \tag{3.55} $$

Similarly, ${}^{M}\boldsymbol{A}_{a}$ is calculated as:

$$ {}^{M}\boldsymbol{A}_{a} = \begin{bmatrix} {}^{M}\hat{\boldsymbol{x}}_{a} & {}^{M}\hat{\boldsymbol{y}}_{a} & {}^{M}\hat{\boldsymbol{z}}_{a} & {}^{M}\hat{\boldsymbol{p}}_{A} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.56} $$

where ${}^{M}\hat{\boldsymbol{x}}_{a}$, ${}^{M}\hat{\boldsymbol{y}}_{a}$ and ${}^{M}\hat{\boldsymbol{z}}_{a}$ are the unit vectors of the auxiliary frame expressed in the MiR reference system, obtained through:

$$ {}^{M}\hat{\boldsymbol{x}}_{a} = \frac{{}^{M}\boldsymbol{p}_{B} - {}^{M}\boldsymbol{p}_{A}}{\|{}^{M}\boldsymbol{p}_{B} - {}^{M}\boldsymbol{p}_{A}\|} \tag{3.57} $$

$$ {}^{M}\hat{\boldsymbol{z}}_{a} = \frac{\left({}^{M}\boldsymbol{p}_{B} - {}^{M}\boldsymbol{p}_{A}\right) \times \left({}^{M}\boldsymbol{p}_{C} - {}^{M}\boldsymbol{p}_{A}\right)}{\|\,({}^{M}\boldsymbol{p}_{B} - {}^{M}\boldsymbol{p}_{A}) \times ({}^{M}\boldsymbol{p}_{C} - {}^{M}\boldsymbol{p}_{A})\,\|} \tag{3.58} $$

$$ {}^{M}\hat{\boldsymbol{z}}_{a} = {}^{M}\hat{\boldsymbol{z}}_{a} \times {}^{M}\hat{\boldsymbol{x}}_{a} \tag{3.59} $$

Finally, for ease of visualization of results, the data are compared in the world reference system, figure 3.27, obtained by rotating the Optitrack reference system by $-\pi/2$ around x_p. This results in a frame with its origin within the test area and its xy plane parallel to the ground. Specifically, the transformation matrix between the Optitrack system and the world is explicitly given by:

$$\boldsymbol{A}_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.60}$$

To transform the data from the MiR system into the world system, the matrix:

$$\boldsymbol{A}_M = \boldsymbol{A}_o {}^o\boldsymbol{A}_M \tag{3.61}$$

is used. In practice, Optitrack markers are placed on the upper face of the box on which the UR5 base is mounted. The markers are arranged to create an asymmetrical geometry, easily recognized by Optitrack. These markers define a rigid body to which the MiR pose is associated. In particular, the position of the MiR is defined by the marker that, considering the projection of the robot onto the support surface, identifies the center line of the drive wheels, figure 3.24. Through its navigation interface, the MiR is then moved to three different points 3.26. The positions measured by Optitrack will be ${}^p\boldsymbol{p}_A$, ${}^p\boldsymbol{p}_B$, and ${}^p\boldsymbol{p}_C$, while those acquired from the MiR feedback data correspond to ${}^M\boldsymbol{p}_A$, ${}^M\boldsymbol{p}_B$, and ${}^M\boldsymbol{p}_C$. Finally, ${}^o\boldsymbol{A}_M$ is calculated using equations $3.51 - 3.59$.

Experimental trajectory data acquisition

For the identification and validation of the model of the MiR250, two types of trajectories for the mobile robot were considered. The first class of trajectories is identified with the label I and is characterized by nearly straight paths. The second family of trajectories consists of curved paths and is referred to with the label C. Operationally, the start and end points of the mission have been appropriately chosen using the navigation software. To obtain the C trajectories, inaccessible areas were set up to guide the MiR along a curve, as shown in figure 3.28, using its control software. During this phase, the UR5 robot is stationary. Table 3.3 reports the

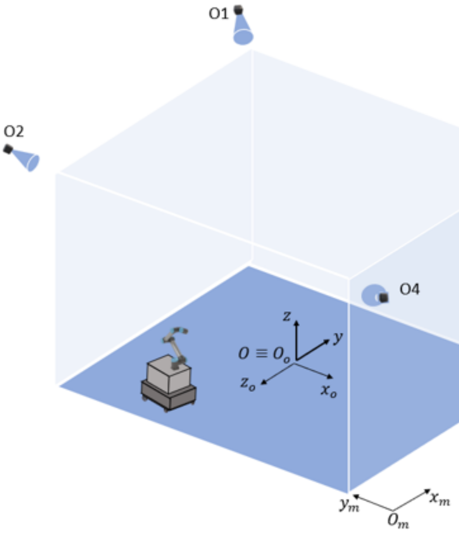


Fig. 3.27 Experimental setup with the reference frames.

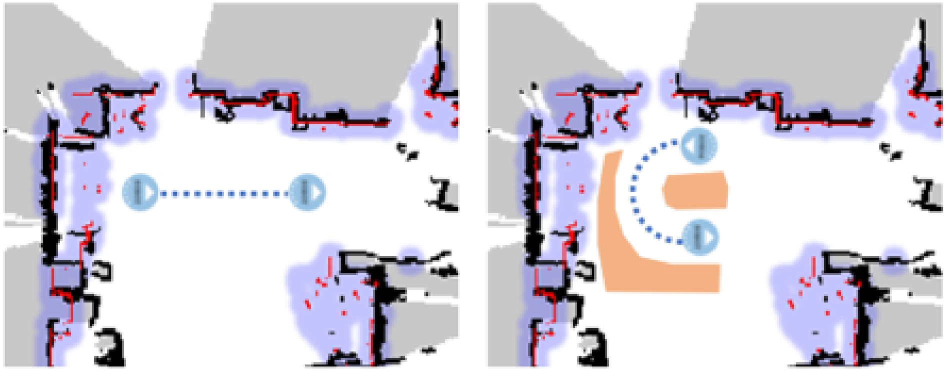


Fig. 3.28 Trajectories for model identification and validation, as seen in the MiR software.

acquisitions that have been used in the model optimization and validation phase. For the tests, the three types of trajectory have been saved in the software and then imposed to the robot when needed, in order to obtain always the same movements. The maximum linear velocity imposed through the software is $1.0m/s$. In the circular movement, the robot runs a curve having $1.25m$ radius. Two examples of trajectories are shown in figure 3.29. For each class of tests, the following data are acquired:

- Trajectory measured by Optitrack, in terms of position and orientation.
- Feedback trajectory from MiR, in terms of position and orientation.
- Feedback velocity from MiR.

Table 3.3 Trajectories for MiR model identification and validation.

Test	Trajectory Type	UR motion (Y=yes, N=No)	Purpose
IN1	I	N	Estimation
IN2	I	N	Estimation
IN3	I	N	Estimation
CN1	C	N	Validation
CN2	C	N	Validation

- Set velocity from MiR.

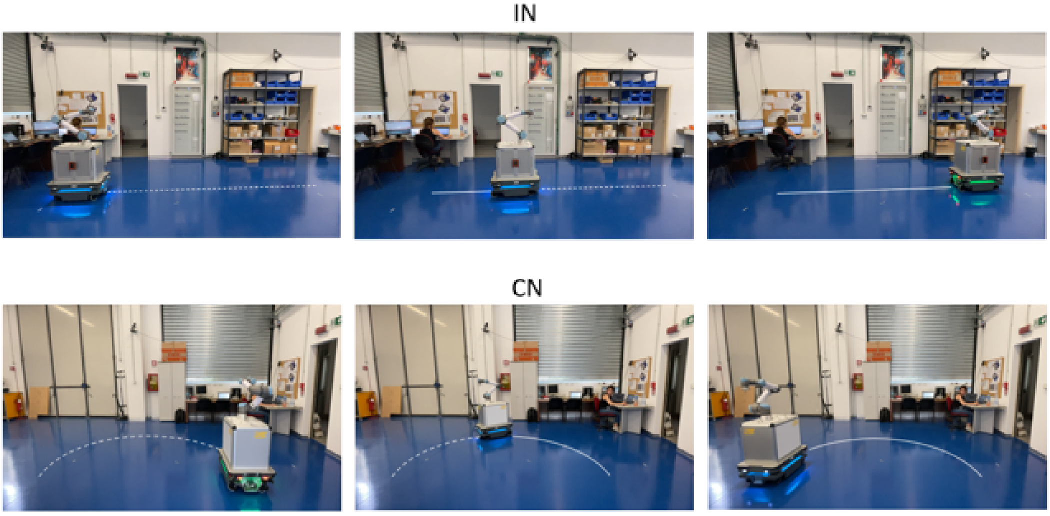


Fig. 3.29 MiR moving during experimental acquisitions.

The experimental tests for the validation of the complete robotic system model are analogous to those for the study of the mobile robot alone. The difference is that the tests were conducted moving the UR5 robot, too, in order to generate a disturbance on the MiR and verify that this disturbance is appreciable on the integrated model. Additionally, the trajectories of the robotic arm end-effector were measured using Optitrack by placing markers on the last link, as shown in figure 3.30. The experimental trajectories of the MiR and UR5 have been compared with those of the model for validation. Also in this case, nearly straight trajectories and curved trajectories of the MiR were considered. Meanwhile, the UR5 robot is positioned overhanging the support surface and actuated by moving the first joint, generating curved trajectories of the end-effector, figure 3.30. The motion of the UR5 robot is generated using Polyscope, the embedded software of the robot. The first joint motion over time can be seen in the bottom plot in figure 3.38. To synchronize the

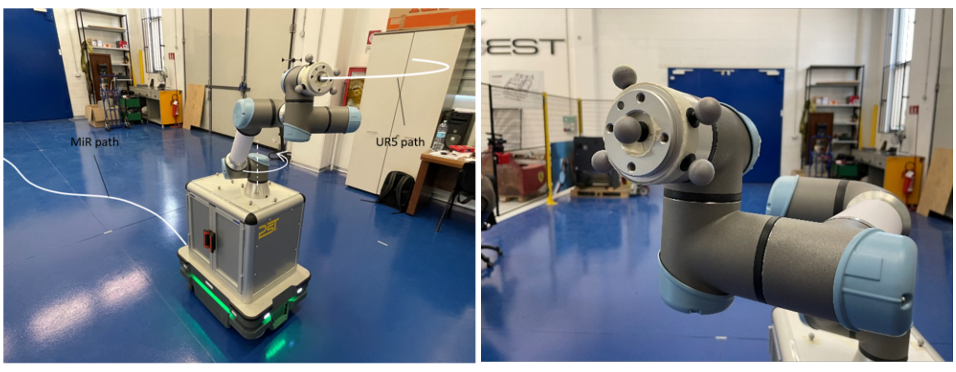


Fig. 3.30 Markers on the end-effector of the UR5.

movement of the UR5 robot with the time interval of the MiR mission, a program written in the UR5 software actuates the robotic arm upon receiving a digital trigger from the MiR controller. Table 3.4 reports the tests used for validation, while figure 3.31 shows two examples of the trajectories considered.

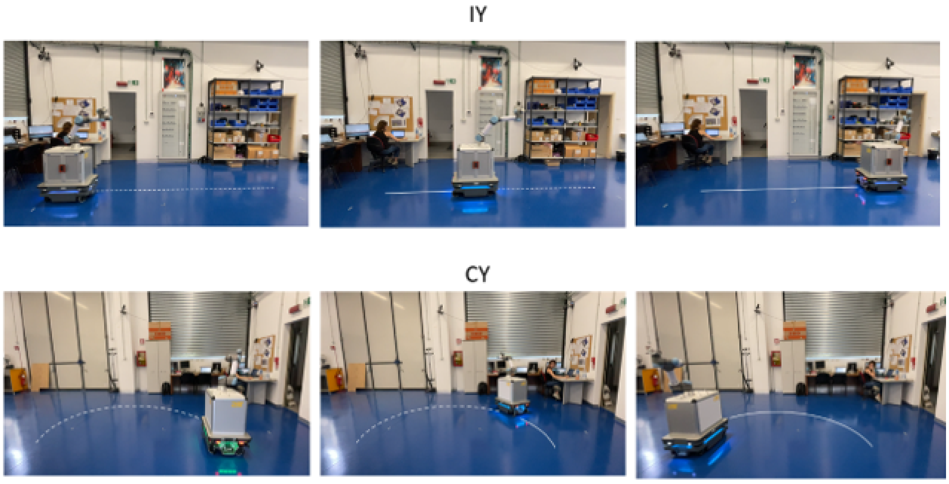


Fig. 3.31 MiR and UR5 moving during experimental acquisitions.

Table 3.4 Trajectories for robotic system identification and validation.

Test	Trajectory Type	UR motion (Y=yes, N=no)	Purpose
IY1	I	Y	Validation
IY2	I	Y	Validation
CY1	C	Y	Validation

3.5.3 Model identification and Validation

Once the model of the robotic system has been formulated, its parameters need to be defined. Some parameters are available from the datasheet, others have to be estimated.

The first step consists in the identification of the model of the MiR robot. The drawing of the parts have been used to estimate their dimensions. The mass values have been inferred from the geometry of the available drawings, and an initial estimate was chosen based on these. In table 3.5 the data used as first try in the model are listed.

Table 3.5 Initial Data of MiR250 Model

Available Data	
Total Mass MiR250 [kg]	83
Height MiR [m]	0.300
Length MiR [m]	0.800
Width MiR [m]	0.580
Driving wheel diameter [m]	0.200
Caster wheel diameter [m]	0.125
Driving wheel axis, along y [m]	0.410
Caster wheel axis, along y [m]	0.375
Caster wheel axis, along x [m]	0.500
Motoreducer Reduction ratio [-]	10.167
Nominal current of the motor [A]	20.7
Torque constant of the motor [Nm/A]	0.15
Estimated Data	
Wheel density [kg/m^3]	100
Wheel joint damping [N*m/(rad/s)]	1e-5
Friction coefficient f [-]	0.3
Friction coefficient f_a [-]	0.5
Slippage s_1 [-]	0.2
Slippage s_2 [-]	0.8
Longitudinal Friction force $F_{\lambda,st}$ [N]	122.01
Longitudinal Friction force $F_{\lambda,inst}$ [N]	81.34
Slip angle $a_1 = a_{m,1}$ [rad]	0.0873
Slip angle $a_2 = a_{m,2}$ [rad]	0.4363

Lateral Friction force $F_{\mu,st}$ [N]	97.61
Lateral Friction force $F_{\mu,inst}$ [N]	65.07
Self-Aligning torque M_st [Nm]	1.04
Motor nominal velocity [rad/s]	250
Motor nominal torque [Nm]	2.25
Motor circuit time constant [s]	1e-3
Motor mass [kg]	3
Rotor mass [kg]	1.5
Rotor diameter [m]	0.100
Current loop crossover frequency [rad/s]	2000π
Current loop PID Proportional	20
Current loop PID Integrative	5
Current loop PID Derivative	0
Velocity loop crossover frequency [rad/s]	200π
Velocity loop PID Proportional	20
Velocity loop PID Integrative	0.01
Velocity loop PID Derivative	0
Motor maximum acceleration [rad/s^2]	1822
Velocity loop phase margin [°]	87

For the identification of the model of the MiR, the UR5 have been considered mounted on the robot, but fixed in its configuration.
To enable a comparison between the actual trajectory and the simulated one, it is essential that the velocity set be the same for both the model and the real system. Consequently, the model was assigned the same velocity set that was given to the actual robotic system during the acquisition of the experimental trajectory. The experimental velocity sets were directly obtained from the MiR through the Rest API, and each of them is associated with an experimental trajectory acquired through the Optitrack motion tracking. To identify the model based on experimental data, the Parameter Estimator provided in the Simulink Design Optimization Toolbox was used. The optimization problem is built on the basis of tests IN1, IN2, and IN3, as already said. For each test, the model receives as input the velocity set acquired experimentally from the MiR, and the result is compared with the experimental trajectory. The cost function is defined by the Parameter Estimator, considering

the error between the model trajectory and the Optitrack measurements, in terms of position (x, y) and orientation (φ). First, a sensitivity analysis was performed to identify which of the estimated parameters had the greatest impact on the model response. The sensitivity analysis is conducted by generating random values according to a uniform distribution within a range compatible with the initial estimate. Specifically, 20 different sets of the parameters in the table 3.5 have been considered, and the resulting error was calculated for each set. Through statistical analysis, the most influential parameters have been identified. The statistical index, *Parameter Influence*, is a value between -1 and 1. The higher the value the higher the extent the parameter affects the model response compared to the experimental data. The sign indicates whether an increase in the parameter corresponds to an increase or a decrease in the error. It is observed that for each test, the four most influential parameters are the friction coefficients f and f_a, the control constant $K_{i,i}$ that is the proportional coefficient of the current loop PID, and the rotor mass m_{rotor}. Therefore, tuning is performed on these four parameters due to the negligible influence of the remaining ones. This approach significantly reduces computation times.
The results of the parameter identification are presented in figure 3.32, 3.33 and 3.34, while the results found are in table 3.6. For each test, the velocity inputs are given to the MiR model, and the parameter estimator varies the indicated parameters in order to minimize a cost function, that is the sum of the residual squares of the provided tests, i.e., IN1, IN2 and IN3. As can be seen in the figures, the model has good performances for what concerns the x direction of the trajectory, while a quite big difference still exists in the y direction after different fitting attempts. The improvement is evident, instead, in the orientation φ. Then, the model have been validated comparing the experimental trajectories C1 and C2 with those obtained by the model, in which the velocity inputs are those read from the MiR robot. The results are shown in figure 3.35 and 3.36. Here too, the model does not replicate exactly the MiR robot behavior, but the result is significant, given the complexity of the system.

Table 3.6 Identification parameters.

Parameter	**Initial Guess**	**Identification**
Friction coefficient f [-]	0.30	0.48
Friction coefficient f_a [-]	0.50	0.50
Current loop PID Integrative constant K_{i_i}	5.0	5.5
Rotor motor mass m_{rotor}	1.5 kg	1.62 kg

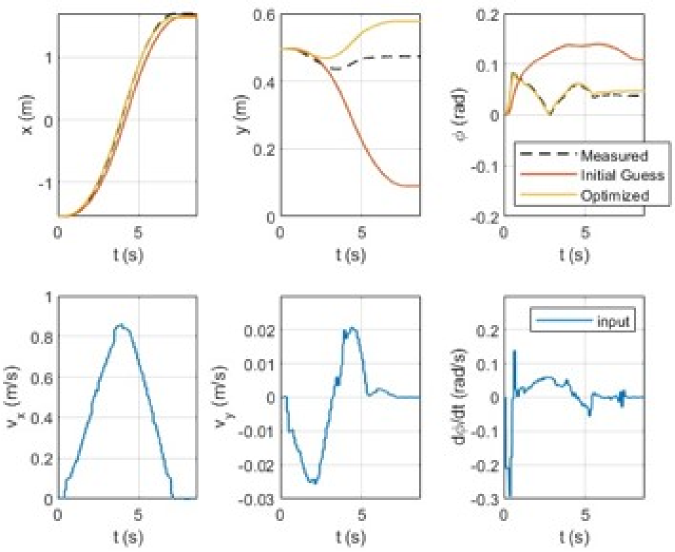


Fig. 3.32 Optimization result - trajectory IN1.

To validate the complete robotic system model, the MiR has been simulated using the velocity set obtained from experimental data, while the robotic arm has been driven using the joint space trajectories acquired from the UR5. In this way, the reaction at the base of the UR5, which constitutes a disturbance for the MiR control, are the same as in the laboratory tests. This allows for evaluating whether the behavior of the mobile base, in the presence of disturbances caused by the collaborative robot, is comparable to the experimental results. The initial validation tests are conducted with IY trajectories. Figure 3.37 illustrates the MiR trajectory obtained from the simulation of the IY1 test. The curves of y and φ show the influence of the motion of the UR5 on the trajectory chasing, but with a slight error with respect to experimental data. The motion of the UR5, both experimental and simulated, is shown in figure 3.38. The motion of UR5 is given only by the first joint, so only the first joint angular position q_1 is reported. In the figure, the data measured by Optitrack and simulated are shown. Where the dot line is not present it means that the end-effector was moving outside the volume captured by the cameras.

The results of the IY2 test are shown in figure 3.39 and figure 3.40. The x and φ components follow the trend of the experimental data. The y component is affected by the oscillation of the UR5 but deviates with the same error observed in the mobile base validation, as expected.

In the CY1 test, the results of which are shown in figure 3.41 and figure 3.42, a

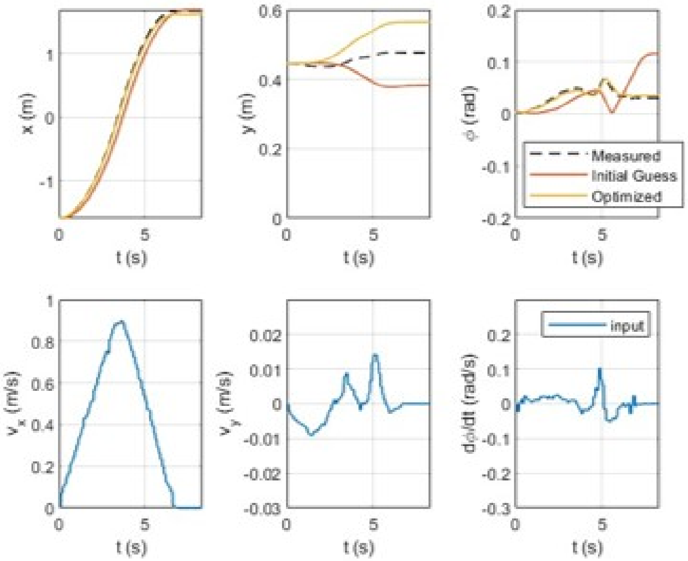


Fig. 3.33 Optimization result - trajectory IN2.

different behavior of the mobile base is observed. In this case, it does not seem to be affected by the movement of the UR5. The oscillatory nature in x, y and φ is not evident. The results are consistent with the experimental findings.

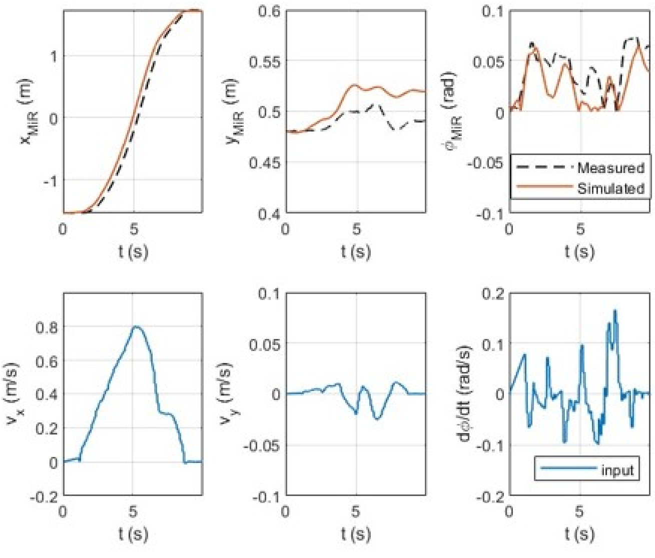


Fig. 3.37 Validation result - trajectory IY1 of MiR robot.

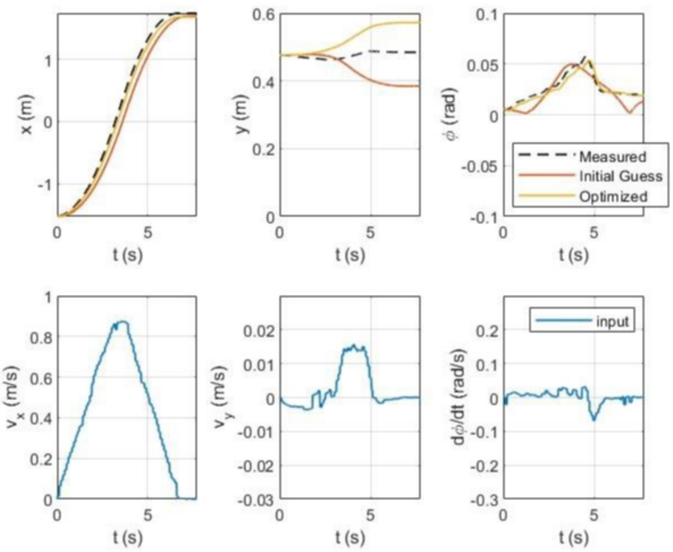


Fig. 3.34 Optimization result - trajectory IN3.

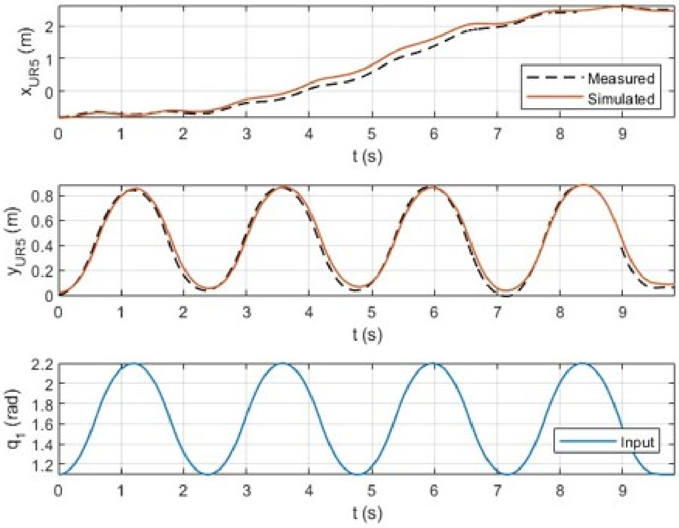


Fig. 3.38 Validation result - trajectory IY1 of UR5 robot.

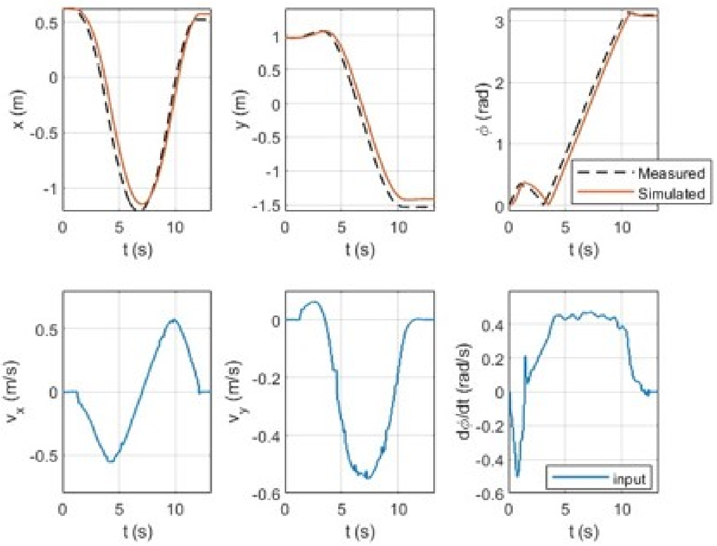


Fig. 3.35 Validation result - trajectory CN1.

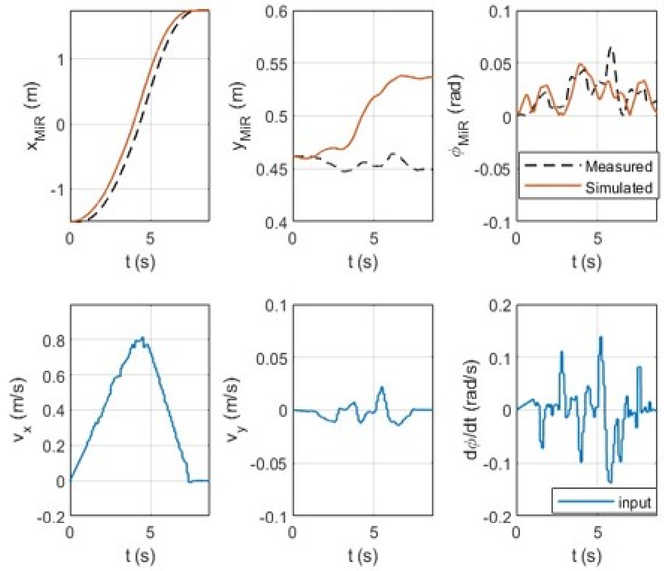


Fig. 3.39 Validation result - trajectory IY1 of MiR robot.

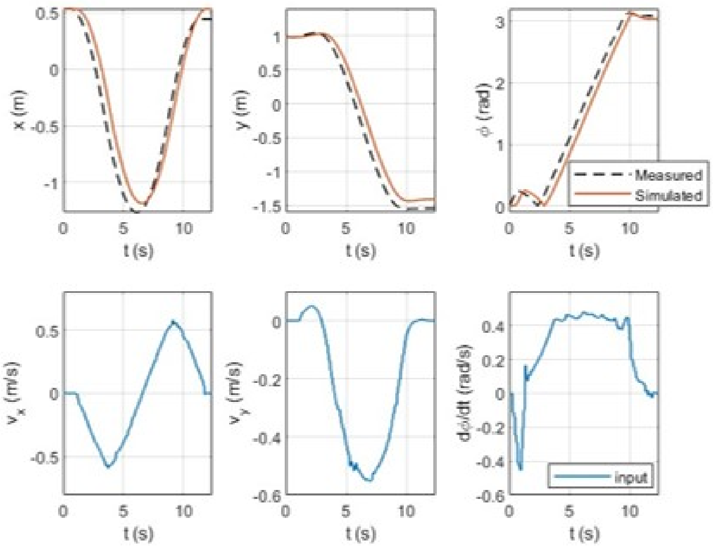


Fig. 3.36 Validation result - trajectory CN2.

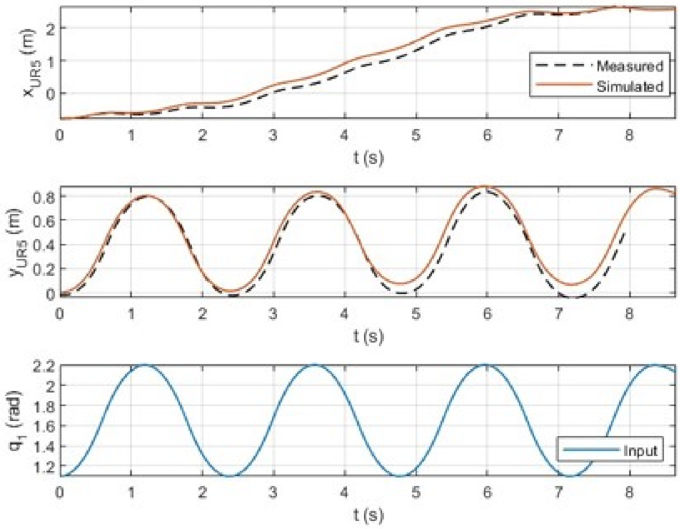


Fig. 3.40 Validation result - trajectory IY1 of UR5 robot.

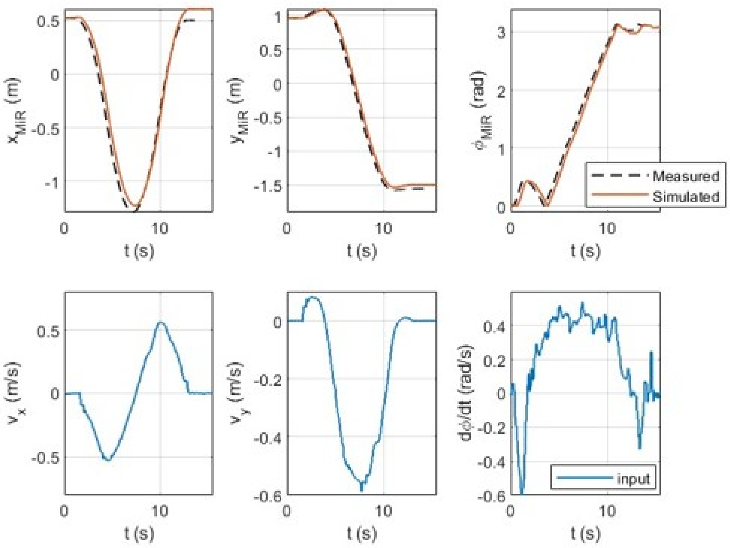


Fig. 3.41 Validation result - trajectory IY1 of MiR robot.

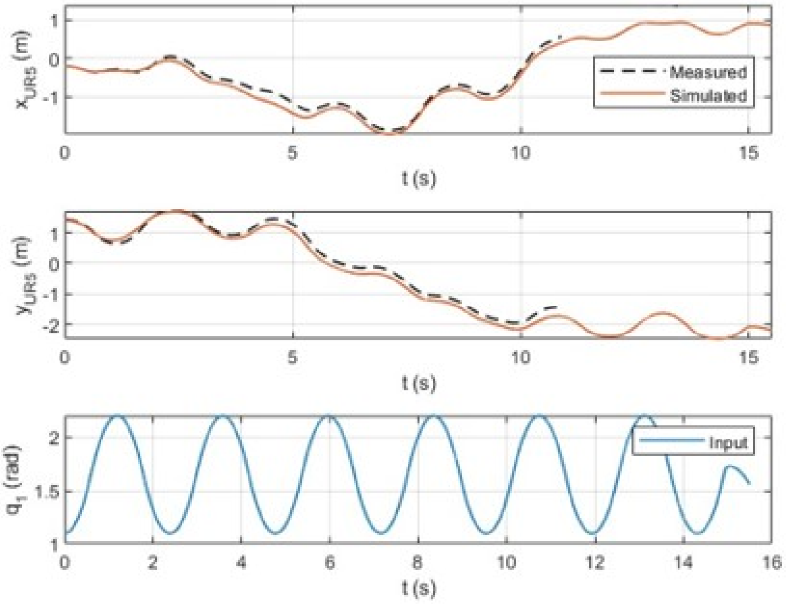


Fig. 3.42 Validation result - trajectory IY1 of UR5 robot.

3.6 Conclusions

The dynamic model of a robotic manipulator has been described. In particular, the focus was to detail the modeling of a mobile robot, that comprises four caster wheels. The contact between wheels and floor is modeled according to Pacejka model. The work was followed by an experimental phase, in which data are collected from the real robots. Then, using the experimental data, an identification phase follows, in order to obtain simulation results as close as possible to the experimental data. The model was then further complicated by adding the disturbance resulting from the dynamics of the UR5, and the study concluded with the validation of the complete MiR-UR5 robotic system model. The following improvements can be made to the model and the identification phase:

- Assume different friction coefficients f and f_a, slippages s_1 and s_2 and slip angles a_1 and a_2 for each wheel. This would significantly increase the model adaptability to the real system.
- Optimize the viscous friction coefficients c on each caster wheel since they were assumed.
- Know the initial position of the caster wheels $\varphi_i(t=0)$, for $i = 1,2,5,6$, to leverage the dynamics of the initial transient for better parameter estimation during the identification phase.

Through a validated model, it is possible to study and improve the system performance. For instance, feed-forward control blocks can be integrated to compensate for trajectory errors due to slipping, or sensor data from the robots can be integrated to diagnose potential malfunctions and ensure the robot's proper functioning.

Chapter 4

Collaborative robotics

In the field of collaborative robotics, the integration of vision systems has revolutionized the way robots interact with their human counterparts. As we advance towards more sophisticated and autonomous robotic systems, the ability of robots to perceive and interpret their environment with the aid of vision systems becomes paramount. Vision systems serve as the eyes of collaborative robots, enabling them to comprehend complex and dynamic environments. By leveraging advanced image processing and machine learning techniques, these systems can detect, recognize, and track objects and human actions. The accuracy of human tracking is crucial for responsive collaboration, as it is essential to ensure safety. Erroneous measurements of human position can lead to unanticipated robot behaviors, posing significant hazards to operators. To mitigate the risk of injuries, some studies have explored the use of wearable devices, which incorporate inertial sensors or active markers. However, these have the drawback to be intrusive or inconvenient for the users. So, in this work, only cameras are considered for tracking the environment the robot moves in. The algorithms discussed in this chapter are designed to address key challenges in human-robot interaction, for preventing collisions and ensuring smooth collaboration.

4.1 Gaze tracking

Nonverbal communication, encompassing eye contact, gestures, and body language, plays a pivotal role in human interactions, particularly during the process. These nonverbal cues are critical for coordinating actions and ensuring a seamless interaction process. Gaze tracking has been investigated as an instrument that can enhance human-robot collaboration.

4.1.1 Use of gaze tracking

Gaze tracking, also known as eye tracking, is a technology that monitors and records the direction and movement of a person's gaze. It determines where a person is looking, for how long, and in what sequence, trying to understand person needs and desires [76]. Gaze tracking systems use various methods, categorized into invasive and non-invasive techniques. Invasive systems typically use wearable devices like specialized glasses equipped with sensors to monitor eye movements. These devices offer high accuracy but can be intrusive and uncomfortable, affecting natural behavior. Non-invasive systems, on the other hand, utilize cameras positioned to capture eye movements without physical contact. These systems often employ infrared light to illuminate the eyes and detect reflections from the cornea and retina. While less intrusive, they may be less accurate in uncontrolled environments.

In collaborative robotics, gaze tracking is essential for improving interaction and safety. By monitoring the operator's gaze, robots can predict intentions, improve task coordination, and enhance safety by identifying potential collision areas. This allows robots to anticipate actions, synchronize movements, and avoid accidents, leading to more efficient and safe human-robot collaboration. [77, 78] confirms that gaze orientation can predict walking trajectories in advance and accurately, providing valuable insights for designing socially-aware navigation systems for robots. Recent studies have demonstrated the practical applications and benefits of gaze tracking in human-robot collaboration. For instance, [79] highlights the effectiveness of using gaze tracking for recognizing human intentions in collaborative assembly tasks. By employing head-mounted displays integrated with gaze tracking systems, the study achieved comparable performance to traditional wearable sensors in terms of user preference and classification time. This approach allows robots to better synchronize their actions with human operators, enhancing overall collaboration efficiency and safety. [80] investigates the use of gaze control versus hand control for telepresence robots. The findings indicate that while gaze control is feasible especially for people with limited motion capabilities. [81] demonstrates that by using face orientation to indicate the intended direction and adhering to the principles of proxemics in a manner similar to humans, the robot is able to execute avoidance maneuvers that

appeared more human-like.
The primary objectives of studying gaze tracking in collaborative robotics are for two aims:

- Provide early information to the robot on where to work: by tracking the operator's gaze, the robot can receive early cues about the intended work area. This allows the robot to preemptively move to or prepare for operations in that specific area, thereby improving efficiency and responsiveness.
- Predict possible collision areas by tracking operator movements: gaze tracking can also be used to predict where the operator is likely to work next. This predictive capability is crucial for collision avoidance, as it enables the robot to foresee potential interaction zones and adjust its path or actions accordingly to avoid collisions.

By leveraging these gaze tracking capabilities, the aim is to enhance the safety and effectiveness of human-robot collaboration, ensuring smoother and more intuitive interactions in shared workspaces.

4.1.2 Gaze-tracking systems

Traditionally, gaze tracking methods have predominantly utilized intrusive techniques, requiring physical contact through devices such as contact lenses, electrodes, or head-mounted apparatuses. Conversely, non-intrusive or remote approaches primarily depend on vision-based methods, employing cameras to capture images of the eyes [82]. Intrusive approaches necessitate direct contact, often resulting in user discomfort and interference. As technology advances towards intelligent systems, the emphasis has shifted from solely accuracy to also enhancing user experience. Consequently, non-intrusive gaze tracking systems, known as remote eye gaze trackers (REGT), have become the preferred choice. The distinction between intrusive and remote systems highlights the inherent trade-offs between precision and user convenience. While intrusive methods excel in accuracy, they can be inconvenient for users in certain contexts. Conversely, remote eye trackers, though generally less precise, provide unobtrusive solutions that are advantageous in scenarios where user comfort and mobility are prioritized.
In REGT, two software techniques can be distinguished: feature-based methods and

appearance-based methods. Feature-based method estimates gaze tracking using extracted local features such as contours, eye corners, and reflections from the eye image. These techniques identify specific features within the eye region by employing machine vision techniques. Detecting elements such as the pupil and glints is relatively straightforward in these methods, particularly when using active light models. Feature-based method is the predominant approach for gaze estimation [83]. In contrast, appearance-based methods focus on the overall appearance of the eye region rather than specific features. These approaches employ machine learning or deep learning algorithms to analyze attributes such as eye color, texture, and pixel intensity patterns. During the training phase, subjects look at predefined locations on a screen, providing data that trains the model to link specific eye appearances with gaze directions. This learned data allows the model to predict a person's gaze based on the overall appearance patterns of their eyes. Algorithms like genetic algorithms, Bayesian classifiers, support vector machines, and artificial neural networks can be used for feature extraction and gaze point mapping [83]. Despite significant research studies, gaze tracking is still challenging due to factors such as eyelid occlusion, variations in eye size, reflectivity, and head pose. These issues add complexity to the task, and as a result, no single affordable solution can address all these challenges comprehensively. However, numerous devices and software solutions are available for gaze tracking, each with its own strengths and limitations.
Both open-source solutions have been investigated:

- GazeParser [84]: GazeParser is an open-source library for low-cost gaze tracking and data analysis. Despite its good accuracy in determining gaze direction, it requires for a head and chin rest to function optimally.

- EyeTab [85]: EyeTab is designed for portable devices like tablets, eliminating the need for external hardware, such as cameras or infrared illuminators, thus ensuring a non-intrusive user experience. However, it works optimally at a fixed distance of 20 cm.

- OpenGazer [86]: OpenGazer requires a webcam and only operates on Linux. Additionally, its accuracy might be limited due to exclusive support for webcams.

- OpenEyes [87]: OpenEyes is software that allows eye tracking with both webcams and infrared cameras. It requires MATLAB but does not provide a data analysis option.

- ITU Gaze Tracker [88]: ITU Gaze Tracker is tailored to work with webcams with infrared lighting. Written in C#, it necessitates some technical proficiency for configuration. Its primary function involves eye-controlled cursor navigation, often used in conjunction with a typing application for visual typing.

- GazeTracking [89]: GazeTracking is a Python library that provides a webcam-based eye tracking system. It gives the exact position of the pupils and gaze direction in real-time.

- OpenFace 2.0 [90, 91]: OpenFace 2.0 is an open-source tool capable of facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. It includes source code for both running and training the models. The output includes these detections and estimations, which can be saved to disk or sent via network in real-time, making it efficient for various applications. Additionally, it can run with a simple webcam.

- GazeSense [92]: GazeSense is an eye tracking software implemented by Eyeware that tracks eye movements and gaze direction in real time without the need for expensive hardware. It can work with 3D cameras or webcams.

Commercial eye tracking systems were also evaluated. They combine both hardware and software components to provide cohesive and efficient eye tracking results. Among them, Tobii [93] is a global leader in eye tracking. The following instruments have been considered:

- Tobii Pro Glasses 3 [94]: Tobii Pro Glasses 3 is a wearable eye tracker designed to capture what the user is viewing while providing robust and accurate eye tracking data. It supports live view of the scene camera video and includes additional data channels that provide comprehensive eye tracking information, including 2D gaze, 3D gaze, gaze origin, gaze direction, and pupil diameter.

- Tobii Pro Spark [95]: Tobii Pro Spark is a compact, high-performance screen-based camera. Sophisticated image processing algorithms identify relevant features, including the eyes and the corneal reflection patterns. Combined with its dedicated software Tobii Pro Lab and Tobii Pro SDK, it allows for recording, analyzing, and visualizing eye tracking data, such as 3D eye coordinates, raw data, and pupil data.

The choice for the system to employ concentrates on GazeTracking, OpenFace 2.0, GazeSense, Tobii Pro Glasses 3 and Tobii Pro Spark. In table 4.1 the characteristics of the gaze-tracking systems are shown. The high cost of the integrated systems makes them unsuitable for a preliminary study of gaze tracking. Conversely, OpenFace 2.0, despite exhibiting a higher error rate, offers an open-source platform, in MATLAB, too. Moreover, it provides a high operating distance, making it the preferred choice.

Table 4.1 Comparison of Gaze Tracking Systems

Gaze Tracking System	Sampling Rate [Hz]	Accuracy	Operating Distance [cm]	Output	Language	Price
Gaze Tracking	Variable	Unknown	30-60	Gaze direction, Pupil position	Python	Free
OpenFace 2.0	Variable	Mean error 9°	30-100	Gaze direction, Eye landmarks	MATLAB, C++, Python	Free
GazeSense	10-90	1.5°	30-80	Gaze direction, Pupil origin	Python, C++	2000€
Tobii Pro Glasses 3	50 or 100	0.6°		2D, 3D gaze, Gaze origin, Gaze direction	Python, Javascript, HTML	20000€
Tobii Pro Spark	33 or 60	0.45°	45-90	Gaze origin, Gaze direction	Python, MATLAB, C	10000€

OpenFace 2.0

OpenFace 2.0 is a facial behavior analysis toolkit designed for computer vision and machine learning researchers. It provides accurate facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation [90]. OpenFace 2.0 uses a pipeline of core technologies for facial behavior analysis, including facial landmark detection and tracking, head pose estimation, eye-gaze tracking, and facial action unit recognition, and can work in real-time. In figure 4.1 the pipeline of the algorithm is shown. OpenFace 2.0 extracts facial landmarks

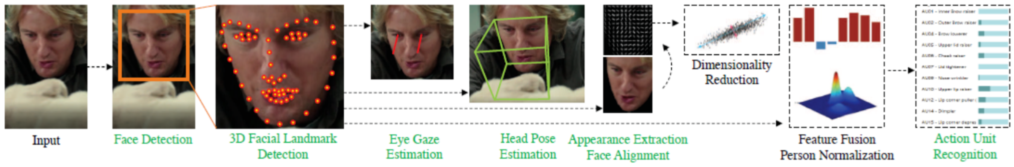


Fig. 4.1 Pipeline of OpenFace 2.0, from [90].

using an advanced methodology known as the Convolutional Experts Constrained Local Model (CE-CLM), which precisely identifies key points such as the eyes, nose, and mouth. The extraction process is made of two main components: the Constrained Local Model (CLM), which uses a statistical model to understand general variations in facial landmark shapes, and the Convolutional Experts (CE), a convolutional neural networks that enhance the recognition of localized variations in facial appearance by focusing on specific regions of the face to improve accuracy. Head pose estimation is achieved by combining the facial landmarks with a 3D face model. For eye gaze estimation, a Constrained Local Neural Field (CLNF) landmark detector is used. This detector identifies key landmarks such as the eyelids, iris, and pupil. The detected pupil and eye location are then used to compute the eye gaze vector for each eye. The computation involves projecting a ray from the camera origin through the center of the pupil in the image plane. The point of intersection of this ray with the eyeball sphere is calculated to determine the pupil's location in 3D camera coordinates. The gaze vector is subsequently estimated as the vector from the 3D eyeball center to the pupil location.

OpenFace 2.0 can be integrated in any C++, C, or MATLAB based project and can work with any camera. The outputs of the algorithm are:

- face_id: indicates if only one (0) or more faces (1) are detected in the frame.
- timestamp: is the elapsed time from the beginning of the video to when the frame was processed, in seconds.
- confidence: denotes the confidence level of the face tracker, typically approximately 0.98.
- success: binary indicator that shows whether the face tracker successfully detected a face in the frame.

- gaze_0_x, gaze_0_y, gaze_0_z: represent the eye gaze direction vector in camera coordinates for eye 0 (the left eye in the image), normalized.
- gaze_1_x, gaze_1_y, gaze_1_z: represent the eye gaze direction vector in camera coordinates for eye 1 (the right eye in the image), normalized.
- gaze_angle_x, gaze_angle_y: indicate the eye gaze direction in radians within camera coordinates, averaged for both eyes. These values provide a more convenient representation than the gaze vectors, with gaze_angle_x indicating horizontal gaze (left-right) and gaze_angle_y indicating vertical gaze (up-down).
- eye landmarks: includes multiple columns capturing the x, y, and z coordinates of 55 individual eye landmarks for each eye.
- Head pose: position and rotation of the head relative to the camera, in 3D space.
- Facial landmarks: landmarks of the entire face for detailed facial analysis.
- Facial action units (AUs): information related to Facial Action Units, which represent facial muscle movements associated with different expressions.

They are saved in a .CSV file, that has as many rows as the frames that are processed by the algorithm. These parameters allow to track the gaze of the user. The technology was applied to the command of a UR5 robot, developing gaze driven trajectories. The system and the experimental results are shown in section 4.3

4.2 Collision avoidance algorithms for customized trajectory

Collision avoidance is a pivotal aspect of collaborative robotic systems, ensuring that robots can navigate their environments without causing harm to humans or damaging objects. This capability is crucial for maintaining safety and efficiency in shared workspace. Collision avoidance algorithms empower robots to detect and react to obstacles in real-time, facilitating seamless human-robot interaction. In this chapter, an innovative collision avoidance algorithm, applied to mobile robotics, is shown. The algorithm allows to condition the trajectory planning according to the preferences of the user, in order to define a predictable path.

4.2.1 State of the art

The development of collision avoidance methods has evolved significantly over the years, incorporating advanced technologies and methodologies to enhance the precision and reliability of these systems. Two scenarios can be considered: a first one in which the ambient is a-priori known, and it is necessary to define the path the robot has to follow, avoiding eventual existing obstacles. This can be done offline, prior to initiating the task. And a second scenario, in which the robot operates without prior information about the spatial constraints of a dynamic environment. In such case, the robot must dynamically adjust its path upon detecting obstacles with its sensors. Moreover, since the obstacles may themselves be in motion, the robot must make swift decisions for evasive maneuvers. These two scenarios correspond to two types of trajectory planning: global methods and local methods [96]. Global methods involve defining the robot entire trajectory from the start point to the goal point and typically require pre-existing knowledge of the environment. Consequently, global methods are usually implemented offline. In contrast, local methods focus on the robot's immediate surroundings, allowing it to adjust its path in response to nearby obstacles. Therefore, local methods are more suited for online collision avoidance, where the robot must react to changes in its environment in real-time. They do not change the complete trajectory, but act according to the input of the robot, so the robot moves according to its surrounding. Among local methods, artificial potential field (APF) based method stands out.

In APF-based methods, both the target and obstacles are represented using potential functions. The overall potential field is designed to exhibit a global minimum at the target location, while high potential values correspond to the positions of obstacles. The gradient of this potential function is employed as the control vector, guiding the robot towards the target while avoiding collisions. Although APF can be utilized for global planning, its most prevalent application is in online local planning. Obstacle avoidance using APF was introduces in [97]. Then, many studies followed. Despite its effectiveness, the APF method is prone to the local minima problem, where the robot can get trapped. To address this, various strategies have been developed. A superquadratic potential function was introduced in [98] to eliminate local minima around obstacles. Additionally, the navigation function concept, described in [99], was proposed to solve local minima issues, although it requires prior information, stationary obstacles, and a fixed destination. Inspired by the limitations of APF, researchers have developed alternative methods such as harmonic potential functions [100], the dynamic window approach [101], the collision cone [102], and attractor dynamics [103]. These methods aim to offer more robust solutions for collision avoidance in dynamic environments.

Another significant direction in potential field research involves practical approaches that reinterpret artificial potentials. For instance, gradient tracking control based on sliding mode, as discussed in [104], considers the gradient as the desired velocity vector field for the robot. This method allows for precise tracking of gradient lines and ensures convergence to the goal, unlike the standard method that requires a dissipative term in the control scheme to ensure asymptotic stabilization [97]. Enhancements to the gradient vector field, such as adding tangential fields, help prevent local minima and create convenient paths. Tangential fields can guide the robot around obstacles, while selective attraction and tangential fields can direct the robot towards the goal from a specific angle [105]. The virtual target approach, where the global target is replaced with a local one around the obstacle, is another method to overcome local minima [106]. Temporary or projected goal positions have also been explored [107]. Most research has focused on potential fields built from a single goal and multiple obstacles, with limited studies on using multiple attractors to navigate through attractive regions and avoid obstacles. An introduction to potential fields with multiple attractors and repulsors is provided in [108], discussing the combination of quadratic or exponential functions to model these fields. However, this work does not offer an optimal strategy for shaping the potential field or distinguish between global and local attractors.

Human-robot collaboration necessitates rapid robot responses, making local methods preferable over global ones. Significant developments in collision avoidance for collaborative robotics often draw from local methods. One early example is the application of APF-based techniques to human-robot interaction, where a collaborative robot adjusts its path in real-time based on proximity to a human detected by a Kinect sensor [109]. This approach uses repulsive velocities as the control vector instead of traditional APF, demonstrating effectiveness in HRC contexts. Other studies, such as [110], extend APF-inspired methods by incorporating the velocity of the danger source into the control vector. This strategy, validated with a collaborative robot equipped with Kinect skeleton tracking, shows promising results. Similarly, [111] adjusts the repulsion effect by varying the size of the operator's bounding volumes, enhancing the robot's ability to avoid fast-moving obstacles. n addition to APF-based methods, recent research explores using local optimization techniques (OP) and Rapidly-exploring Random Trees (RRT) for fast human-robot collision avoidance. The method described in [112] formulates collision-free trajectories using OP, recalculating optimal joint positions at each control step. Although this approach is faster than global methods, it is still slower than APF-based techniques. The improved RRT algorithm presented in [113] can handle moving obstacles, but its average planning time remains a limitation for fast-moving scenarios.

4.2.2 New collision avoidance algorithm

A novel potential field with local attractors method is formulated and tested for mobile robot. In this approach, the global attractor, representing the target, is modeled as a quadratic function, while local attractors serve as inflections in the potential field without forming local minima. These local attractors guide the robot through specific areas, particularly around obstacles, ensuring a predictable collision-free path. The primary motivation for this method was to devise a simple and effective strategy to manage obstacles predictably. Here, "predictable" refers to the ability to predefine the side by which the robot will pass an obstacle, which can be crucial when certain collision-free paths are preferred. For example, when the obstacle is a human, additional challenges arise due to potential emotional reactions. As observed in [114], it is essential for the robot not to block the human path. To achieve this, a virtual force is employed to direct the robot to detour behind the human. Furthermore, socially acceptable pre-collision criteria suggest that selecting a specific side when crossing a human path can enhance the legibility of the robot's motion [115]. In shared environments, legibility is linked to effective locomotion, implying minimal effort for all agents involved [116]. This concept is also relevant in human-robot collaboration, where guiding the robot trajectory toward predictable regions can result in smoother interactions [117].

In this section, APF-based technique enhanced with local attractors is addressed. Consider a robot represented as a point in a two-dimensional Cartesian space. The robot task is to navigate from an initial position $\boldsymbol{x}_i$ to a goal position $\boldsymbol{x}_g$. The goal position can be conceptualized as an attractive potential field U_g, which is modeled using a quadratic function, according to [97]:

$$U_g(\boldsymbol{x}) = \frac{1}{2}\sigma\|\boldsymbol{x} - \boldsymbol{x}_t\|^2 \tag{4.1}$$

in which σ is a positive parameter that determines the intensity of the quadratic function, and $\boldsymbol{x} = [x, y]^T$ is a generic position in the plane. Consider an obstacle in the plane. Generally, the obstacle can have any geometry. Often, it is convenient to approximate objects by composing simple shapes, such as spheres, cylinders, and planes. In other applications, more accurate potential functions may be needed to describe the obstacles. For the purposes of this work, the case of a disc in two dimensions is considered. The obstacle is centered in $\boldsymbol{x}_o$, and has radius R_o. It generates a repulsive potential field U_o which is modeled using an exponential function [108]:

$$U_o = \beta_o e^{-\frac{\gamma_o}{2}\|\boldsymbol{x} - \mathbf{x}_o\|^2} \tag{4.2}$$

in which β_o and γ_o are the peak value and the exponential decay, respectively, of the gaussian built on the obstacle. In figure 4.2(a), the outer circle centered at $\boldsymbol{x}_o$ defines the active region U_o^*. This region is characterized by a circle with radius R_o^*, where the gradient of U_o becomes negligible outside U_o^*. The radius R_o^* can be determined by solving $|\nabla U_o| = s_\varepsilon$, where s_ε is a small positive value referred to as the *zero threshold*:

$$R_o^* = \left[-\frac{1}{\gamma_o} W_{-1}\left(-\frac{s_\varepsilon^2}{2\beta_o^2\gamma_o}\right)\right]^{1/2} \tag{4.3}$$

where W_{-1} is the solution that correspond to the lower branch of the Lambert function. For further explanation, refer to [118]. If only the target and the obstacle are considered, the resulting total potential field U_{go} can be expressed as:

$$U_{go}(\boldsymbol{x}) = U_g(\boldsymbol{x}) + U_o(\boldsymbol{x}) = \frac{1}{2}\sigma\|\boldsymbol{x} - \boldsymbol{x}_g\|^2 + \beta_o e^{-\frac{\gamma_o}{2}\|\boldsymbol{x} - \boldsymbol{x}_o\|^2} \tag{4.4}$$

The robot control law can be defined such that the command vector follows the direction of the negative gradient. Referring to the case of figure 4.2(a), if $\boldsymbol{x}_i$, $\boldsymbol{x}_o$ and $\boldsymbol{x}_g$ are aligned, the robot may potentially become stuck at the classical saddle point [99]. However, this is a limit case: with a slight perturbation in the y direction, the robot can potentially follow either the continuous or the dashed path.

An attractive source is positioned at $\boldsymbol{x}_a$, close to the obstacle, as shown in figure

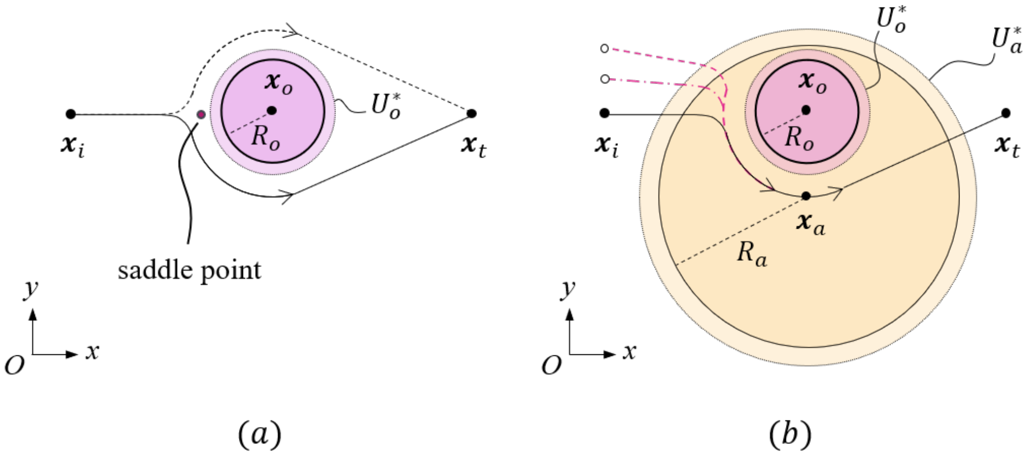


Fig. 4.2 Possible paths when the robot approaches an obstacle. (a) Case without local attractor, and (b) case with local attractor.

4.2(b). The effect of this attractive source is limited to a local area, so it is called local attractor, also to distinguish it from the goal attractive source, which exerts a global effect.

The local attractor potential field U_a is modeled using a negative exponential function [108]:

$$U_a(\boldsymbol{x}) = -\alpha_a e^{-\frac{\gamma_a}{2}\|\boldsymbol{x}-\boldsymbol{x}_a\|^2} \tag{4.5}$$

where α_a and γ_a are positive parameters that regulate the intensity and the decay of the attractive effect, respectively. In figure 4.2(b), a local attractor is shown, centered in $\boldsymbol{x}_a$. The circle having radius R_a^* represents the active region U_a^*, such that the gradient of U_a diminishes to zero outside U_a^*. Tuning α_a and γ_a, it is possible to determine the extent of the active region. The local attractor can include the obstacle, in order to deviate the path of the robot toward the local planner, placed on a preferred side. In 4.2(b), the robot trajectory is affected by a local attractor located on the side of the obstacle. The various line styles depict different possible paths, each associated with distinct initial positions $\boldsymbol{x}_i$. The local attractor effectively influence the robot obstacle avoidance on the side of the local attractor. The active region U_a^* is determined by solving the equation $|\nabla U_a| = s_\varepsilon$:

$$R_a^* = \left[-\frac{1}{\gamma_a} W_{-1}\left(-\frac{s_\varepsilon^2}{2\alpha_a^2\gamma_a}\right)\right]^{1/2} \tag{4.6}$$

The total potential field is then:

$$U_t(\boldsymbol{x}) = U_g(\boldsymbol{x}) + U_o(\boldsymbol{x}) + U_a(\boldsymbol{x}) = \frac{1}{2}\sigma\|\boldsymbol{x}-\boldsymbol{x}_g\|^2 + \beta_o e^{-\frac{\gamma_o}{2}\|\boldsymbol{x}-\boldsymbol{x}_o\|^2} - \alpha_a e^{-\frac{\gamma_a}{2}\|\boldsymbol{x}-\boldsymbol{x}_a\|^2} \tag{4.7}$$

Examples of the total potential field are illustrated in figure 4.3. The contour lines of the potential fields are projected on the planes having $z = 0$. If the parameters of the local attractor are not properly tuned, with the presence of the global attractor, a local minimum may arise, as in figure 4.3(a). There exist values of α_a so that the local minimum is avoided, as in figure 4.3(b). To streamline the analysis and effectively manage the coexistence of attractors, two constraints must be applied. The first constraint ensures that the local attractor is placed at a sufficient distance from the active region of the obstacle potential field U_o^*:

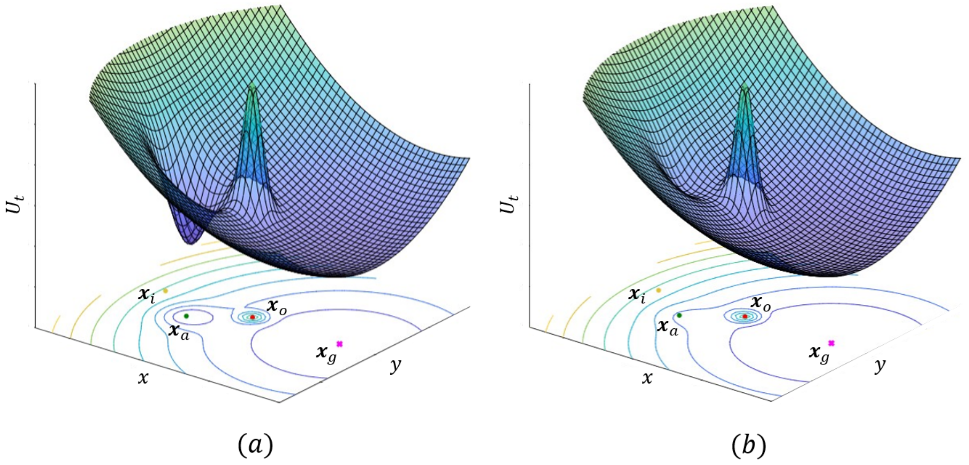


Fig. 4.3 Total potential attraction fields, (a) when the local minimum occurs, and (b) when the local minimum does not occur.

$$\|\boldsymbol{x}_a - \boldsymbol{x}_o\| > R_o^* + \tilde{\varepsilon} \tag{4.8}$$

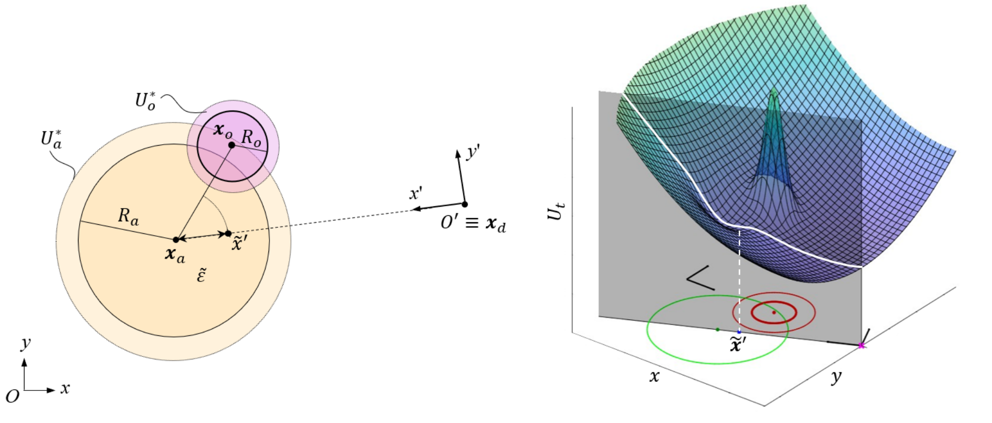
with $\tilde{\varepsilon}$ a distance that depends on the relative position of $\boldsymbol{x}_o$, $\boldsymbol{x}_a$ and $\boldsymbol{x}_g$. The second constraints requires the global minimum to be far from the global attractor, i.e.:

$$\|\boldsymbol{x}_a - \boldsymbol{x}_g\| \geq R_a^* \tag{4.9}$$

The local minimum problem can be addressed by examining the saddle point near the local attractor. As the parameter α_a increases, the total potential starts to curve.

However, if α_a is less than a critical value, denoted as $\tilde{\alpha}_a$, the local minimum does not appear. This limit is defined by the occurrence of a saddle point.

To analyze the saddle point, it is useful to focus exclusively on the potential field U_{ga}, which is composed of U_g and U_a. When constraint 4.8 is met, the saddle point will be located outside U_o^*, where U_t approximately equals U_{ga}. The analysis can be further streamlined by representing the potential $U_{ga}(x', y')$, within an auxiliary reference frame $O' - x'y'$, with $O' \equiv \boldsymbol{x}_o$ and x' - axis aligned with $\boldsymbol{x}_a$, as in figure 4.4. Here, the first partial derivative of the potential field U_{ga} is taken and set to zero.



(a) (b)

Fig. 4.4 Example of obstacle repulsive potential field and local attractive potential field.(a) Sketch of the problem, with reported relevant parameters; (b) representation of the potential field in the same situation.

Given that σ, α_a and γ_a are positive, the saddle point is located on the x' - axis. So, it possible to analyze U_{ga} in $y' = 0$, that is the white curve in figure 4.4(b), formed by the intersection of the surface $U_t \approx U_{ga}$ and the plane $y' = 0$. Consequently, the equation becomes:

$$\frac{\partial}{\partial x'} U_{ga}(x', 0) = \sigma x' + \alpha_a \gamma_a (x' - x'_a)\, e^{-\frac{\gamma_a}{2}(x - x'_a)^2} \tag{4.10}$$

with $\boldsymbol{x}'_a$ being $\boldsymbol{x}_a$ in the auxiliary plane. For small values of α_a, only a single solution exists, that is the global minimum of the equation in $x' = 0$. With the increase of α_a, the resulting potential field bends close to $x' = x'_a$, forming eventually a saddle in $x' = \tilde{x}'$. The corresponding value of α_a, denoted as $\tilde{\alpha}_a$, serves so the upper bond for α_a. For values of α_a exceeding this limit, a local minimum emerges within the range $0 < x' < x'_a$. To determine $\tilde{\alpha}_a$, s, both the first and second derivatives of the potential field must be set to zero. This yields:

$$\tilde{x}'(x'_a,\gamma_a) = \frac{2}{3}x'_a\left[\cos\left(\frac{\theta+4\pi}{3}\right)+1\right] \tag{4.11}$$

$$\theta(x'_a,\gamma_a) = \arccos\left(\frac{27}{2\gamma_a {x'_a}^2}-1\right) \tag{4.12}$$

$$\tilde{\alpha}_a(\sigma,x'_a,\gamma_a) = \frac{-\sigma\tilde{x}'}{\gamma_a\left(\tilde{x}'-x'_a\right)\,e^{-\frac{\gamma_a}{2}\left(\tilde{x}'-x'_a\right)^2}} \tag{4.13}$$

In general, although U_a is centered in $\boldsymbol{x}_a$, the combined effect with U_g results in the saddle point not aligning exactly with $\boldsymbol{x}_a$. Indeed, the saddle point is located at $\tilde{\boldsymbol{x}}' = [\tilde{x}'\ 0]^T$, which is slightly shifted from $\boldsymbol{x}_a$ in the negative direction along x' - axis, figure 4.4(a). This shift can be measured by the distance $||\boldsymbol{x}'_a - \tilde{\boldsymbol{x}}'|| = (x'_a - \tilde{x}') = \varepsilon$. This factor has been accounted for in constraint 4.8 to ensure the saddle point falls outside U^*_o. There are two possible situations:

- The line segment between the attractor and the target position $\overline{\boldsymbol{x}_a\boldsymbol{x}_g}$ does not intersect U^*_o, ensuring that the saddle point is outside U^*_o.
- The line segment $\overline{\boldsymbol{x}_a\boldsymbol{x}_g}$ intersects U^*_o. In this scenario, if constraint 4.8 is not met, the saddle point of U_{ga} may overlap the obstacle, invalidating the approximation $U_t \approx U_{ga}$. Conversely, if constraint 4.8 is met, this issue is avoided.

As a result, the value of $\tilde{\varepsilon}$ in constraint 4.8 for the two cases is:

$$\tilde{\varepsilon} = 0 \qquad if \quad \overline{\boldsymbol{x}_a\boldsymbol{x}_g} \cap U^*_o = 0 \tag{4.14}$$

$$\tilde{\varepsilon} = \varepsilon \qquad if \quad \overline{\boldsymbol{x}_a\boldsymbol{x}_g} \cap U^*_o \neq 0 \tag{4.15}$$

For further details please refer to [118].

The upcoming sections detail the tests conducted to validate the algorithm. For both cases, the experimental setups are first described and then conducted tests are detailed.

4.3 Gaze tracking tests

A gaze-following algorithm that uses gaze tracking for robot control has been validated. Initially, a preliminary validation has been made in simulation environment. Once confirmed the strategy validity and the capability of the robot to follow the gaze input, the algorithm has been tested experimentally.

4.3.1 Experimental setup

The experimental setup comprises a robot, a camera, a PC and a wi-fi router. The PC is used as the external controller for the robot, that handles the data from the camera and the feedback data from the robot, and sends the command back to the robot.
The robot used for the tests is the UR5 CB3-series, from Universal Robots [119]. The Universal Robots UR5 consists of three primary components. The control unit houses the Electrostatic Discharge (ESD) board and includes all ports for cable connections. The teach pendant serves as the interface between the robot and the operator and is used for programming rapidly the robot movements. In figure 4.5, the robot arm, the control unit and the teach pendant are shown. The UR5 itself, which is the robotic arm. The robot control unit enables communication through TCP/IP (Transmission Control Protocol/Internet Protocol). The use of TCP ensures reliable communication between the robot and the PC. The connection is established

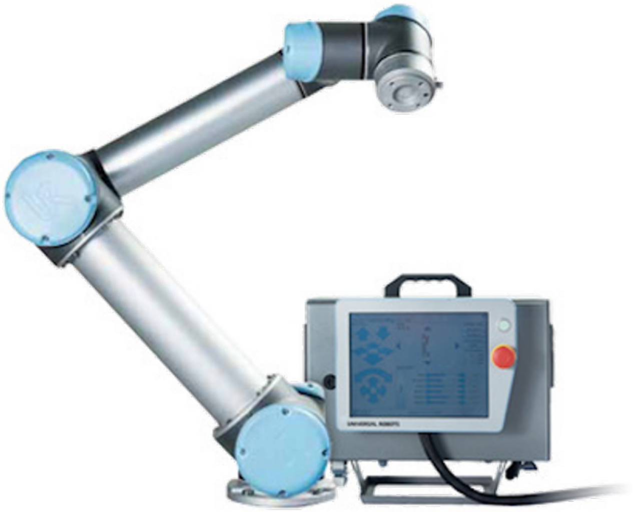


Fig. 4.5 UR5 CB3-series with its control unit and its teach pendant.

via communication sockets, at an operating frequency of 125 Hz.
The robotic arm has six revolute joints forming an open kinematic chain. The UR5 is an anthropomorphic robot with a non-spherical wrist configuration. An end-effector (EE) can be attached to their tool-center-point (TCP) to perform a specific task. In figure 4.6, the robotic arm with the indication of the joints is presented. The first three joints are referred as base, shoulder, and elbow, while the following three ones compose the wrist and are named as wrist 1, wrist 2, and wrist 3. In table 4.2 are listed the techcnical characteristics of the robotic arm, from [120]. The modified DH parameters are listed in table 4.3.

The camera used for the tests is an Orbbec Astra Pro camera [121]. The Orbbec Astra Pro is a high-performance structured light 3D camera. It features advanced depth-sensing technology and a high-resolution RGB camera. The camera operates effectively in various lighting conditions, providing robust performance and ensuring consistent data quality. The Astra Pro offers a depth range of 0.4 to 8 *m* and a

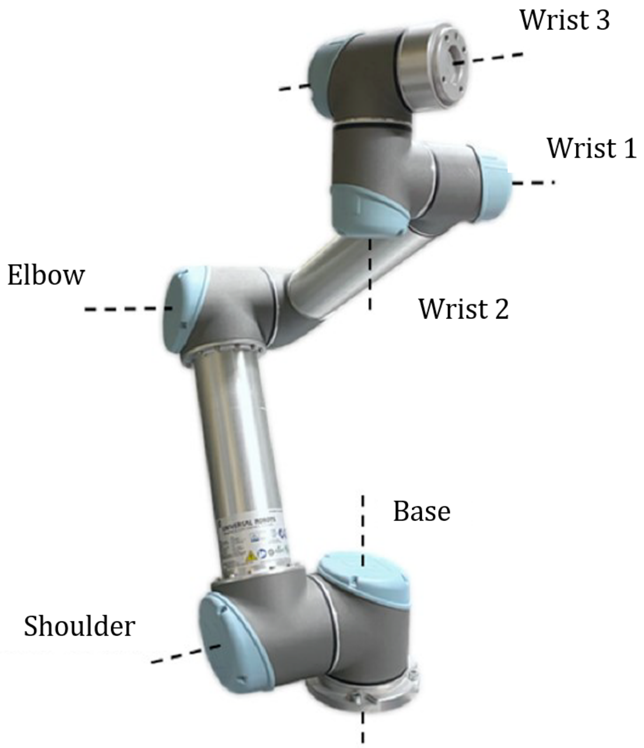


Fig. 4.6 UR5 CB3-series with indications of its revolute joints.

Table 4.2 UR5 Robot Specifications

Specification	Value
Weight	18.4 kg
Maximum payload	5 kg
Reach	850 mm
Joint ranges	±360°
Joint max speed	180 °/s
TCP max speed	1 m/s
Repeatability	±0.1 mm
Degree of freedom	6 revolute joints
Programming	Polyscope graphical user interface
Communication	TCP/IP, Ethernet socket & Modbus TCP

Table 4.3 Modified DH Parameters of the UR5 CB3 series robot.

Joint	θ_{DH} [rad]	a_{DH} [m]	d_{DH} [m]	α_{DH} [rad]
Base	θ_1	0	0.089159	$\pi/2$
Shoulder	θ_2	-0.425	0	0
Elbow	θ_3	-0.39225	0	0
Wrist 1	θ_4	0	0.10915	$\pi/2$
Wrist 2	θ_5	0	0.09465	$-\pi/2$
Wrist 3	θ_6	0	0.0823	0

depth resolution of 640 × 480 pixels at 30 frames per second (fps). Its RGB camera provides a resolution of 1280 × 720 pixels, ensuring high-quality visual data. Additionally, the camera's field of view is 60 horizontally and 49.5° vertically, providing a wide-angle perspective. The Orbbec Astra Pro also features USB 2.0 connectivity, it measures 165 × 30 × 40 *mm* and weights 0.3 *kg*. The camera is powered by the Orbbec OpenNI2 SDK. Before testing, the camera has been calibrated, for tuning camera intrinsic parameters, for better performances.

In figure 4.7, the setup used for gaze-following algorithm tests is shown. The UR5 robot was mounted on a workbench, while the Astra camera was positioned 1.3 *m* away from the robot base. In this configuration, the user is thought to be seated 0.8 *m* away from the camera during the tests.

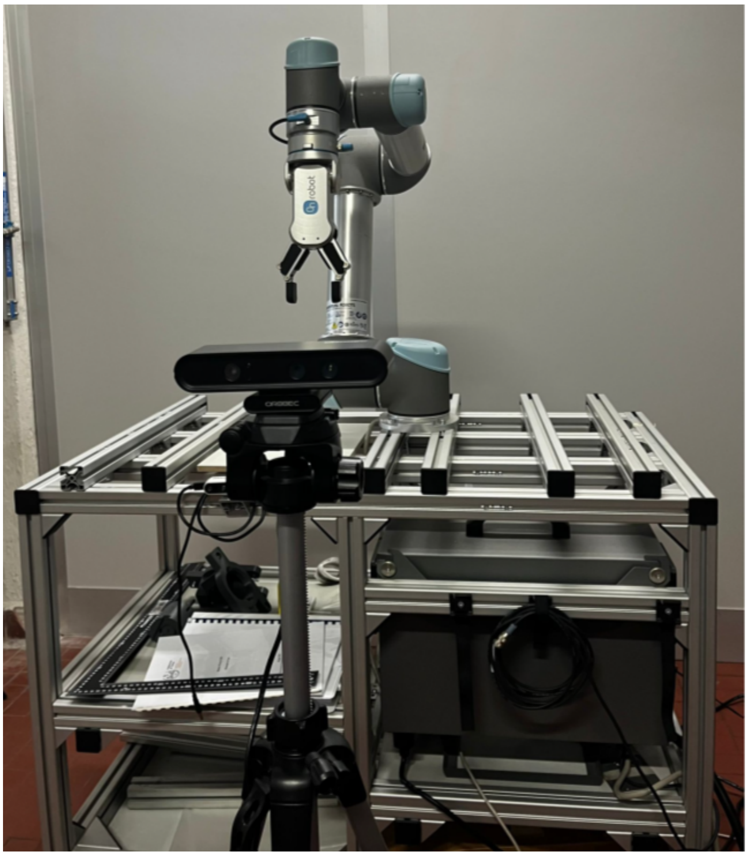


Fig. 4.7 Setup used for gaze-following tests.

4.3.2 Software implementation

The implemented simulation environment using CoppeliaSim and MATLAB is supportive for the experimental tests. CoppeliaSim was utilized for the visualization and interaction with the simulated environment, while MATLAB was employed to process gaze data and control the UR5 robotic arm. Preliminary experiments focused on validating the integration of gaze tracking with CoppeliaSim. These initial tests involved controlling a virtual sphere within CoppeliaSim based on gaze inputs. By successfully controlling the virtual sphere based on gaze direction, these tests confirmed the feasibility of using gaze tracking for dynamic object manipulation in a simulated setting.

To achieve real-time access to crucial gaze tracking data during the experiments, a strategic approach was employed. Recognizing the necessity for immediate access to data rather than waiting for post-execution file retrieval, the solution involved running OpenFace 2.0, specifically the *FeatureExtraction.exe* executable, in the background. The system could collect and process gaze tracking data in real-time without interruptions or delays. Another module in the code is dedicated to retrieving and processing this data. This module operates concurrently with data acquisition, enabling the system to dynamically obtain and process data in real time. The real-time data processing module uses a dual-loop mechanism. The first loop checks if the output CSV file from OpenFace 2.0 has been created correctly and starts reading data from this file as soon as it becomes available. The second loop handles the dynamic retrieval and real-time processing of the data, ensuring that the system seamlessly captures and processes the gaze tracking data generated by OpenFace 2.0. This approach guarantees that the system can provide accurate and timely data for immediate analysis and use. The system extracted and stored the relevant gaze tracking information, processing only the columns of useful information, to speed up the process.

The rest of the code takes the data about gaze tracking, elaborates the robot control and sends instructions to CoppeliaSim.

CoppeliaSim is a highly flexible robot simulation platform, enabling quick and accurate simulations of different physical situations. CoppeliaSim modular architecture allows for the integration of various robotic components and sensors. This modularity supports a wide range of robot types and configurations, from simple robotic arms to complex autonomous vehicles. The platform offers a remote API that allows external applications, including MATLAB, to control and interact with the simulation, enabling users to create custom control algorithms. MATLAB allows integration between OpenFace 2.0 and CoppeliaSim, in order to control the position of a sphere in real-time, driven using the gaze tracking output.

Gaze tracking data, particularly the gaze angles, have been mapped to control the sphere movement. This mapping ensured the accurate translation of changes in the user gaze direction into corresponding movements of the simulated sphere. The precision of this mapping was essential for faithfully simulating the user's visual interactions. The integration established a dynamic feedback loop where the user's gaze influenced the sphere position in real-time. To implement the mapping between gaze angles and sphere movement, the script utilized API functions provided by CoppeliaSim. The function *simxSetObjectPosition* sent position information to CoppeliaSim, adjusting the sphere position based on the mapped gaze angles. The bidirectional gaze control involved simultaneous control along both the x and y axes. This setup allowed the simulated sphere to move freely within the xy plane, responding to changes in both *gaze_angle_x* and *gaze_angle_y*. The bidirectional control experiment involved coordinated variations of the x and y positions of the simulated sphere based on the user's gaze movements. By keeping the z coordinate fixed, the system translated horizontal, *gaze_angle_x*, and vertical *gaze_angle_y*, angular variations into corresponding movements of the sphere within the xy plane in CoppeliaSim. In figure 4.8, the simulated sphere in CoppeliaSim environment is shown. The scene includes only the sphere and a plane for reference, in order to speed up calculation. The mapping between gaze angles and the sphere's coordinates

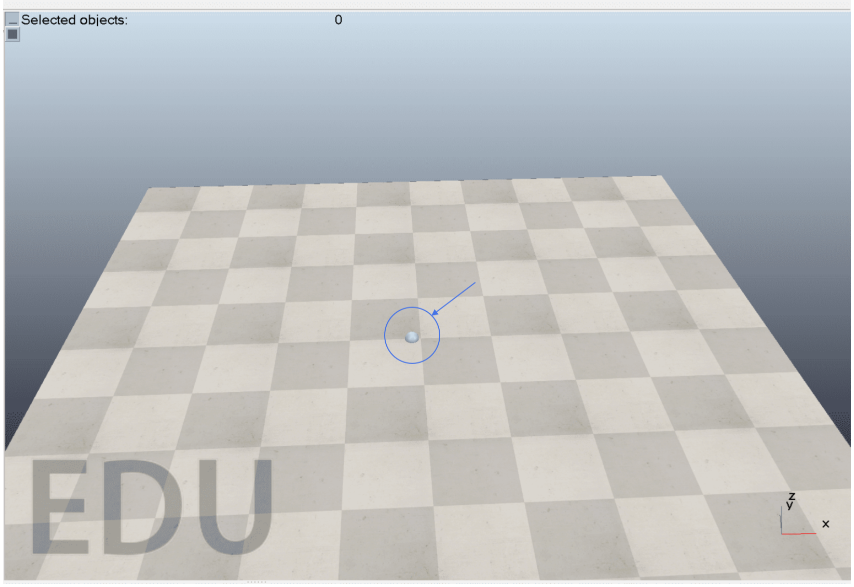


Fig. 4.8 Simulated sphere in CoppeliaSim environment.

was established as follows:

- For the $x - axis$, changes in the horizontal gaze direction, i.e., *gaze_angle_x* were mapped to movements of the sphere along the $x - axis$ in CoppeliaSim. A positive increase in *gaze_angle_x*, that is the gaze shifting to the right, caused the sphere to move to the right, while a negative increase, so the gaze shifting to the left, caused the sphere to move to the left.

- Similarly, for the $y - axis$, changes in the vertical gaze direction *gaze_angle_y* were mapped to movements of the sphere along the $y - axis$ in CoppeliaSim. A positive increase in *gaze_angle_y*, that is gaze shifting downward, caused the sphere to move downward, while a negative increase, that is gaze shifting upward, caused the sphere to move upward.

The setup demonstrated the integration ability to handle complex movements and translate gaze data into sphere control. The successful execution of bidirectional control highlighted the system's effectiveness in integrating multiple gaze tracking parameters concurrently, providing a robust platform for exploring further applications of gaze-driven interactions.

4.3.3 Robot control

The aim is to control robot motion using gaze information. The output is to obtain the robot to follow the position of the operator's point of focus, that has been previously tested moving the simulated sphere.
The workspace of the UR5 is defined as 1 *m*, corresponding to the maximum workspace of the UR5, 850 *mm*, plus the presence of the tool, an OnRobot RG2, [122]. For control sake, two spheres have been defined:

- The workspace Sphere, Vw, centred on the shoulder of the robot. When the gaze enters the the sphere, the robot starts moving towards the target.
- The stopping Volume Vs is essentially the sphere previously simulated, around the gaze point, and allows the robot to stop when the TCP enters this volume.

The robot control strategy is presented inf figure 4.9. The core of the control strategy is the gaze-following algorithm, that is receives gaze information, from the code previously described, and the pose of the robot TCP. The pose is calculated using

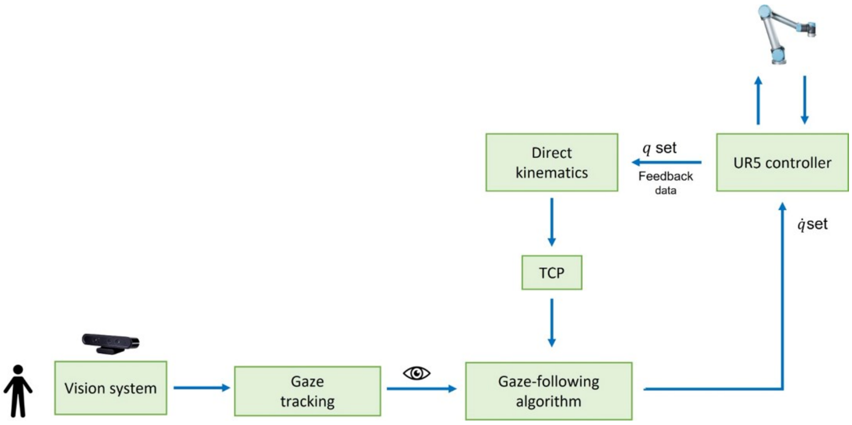


Fig. 4.9 Robot control strategy.

direct kinematics, as the robot returns the joint positions.
The linear velocity of the TCP is determined by the position of the gaze target, taking into account the distance between the TCP and the target. A velocity profile, in operative space, is used, figure 4.10, enabling the TCP to accelerate gradually and move faster when the target is further away, with a maximum velocity capped at 0.35 m/s. The velocity information in operative space is translated to joint velocity information using the Geometric Jacobian matrix of the robot, according to equation:

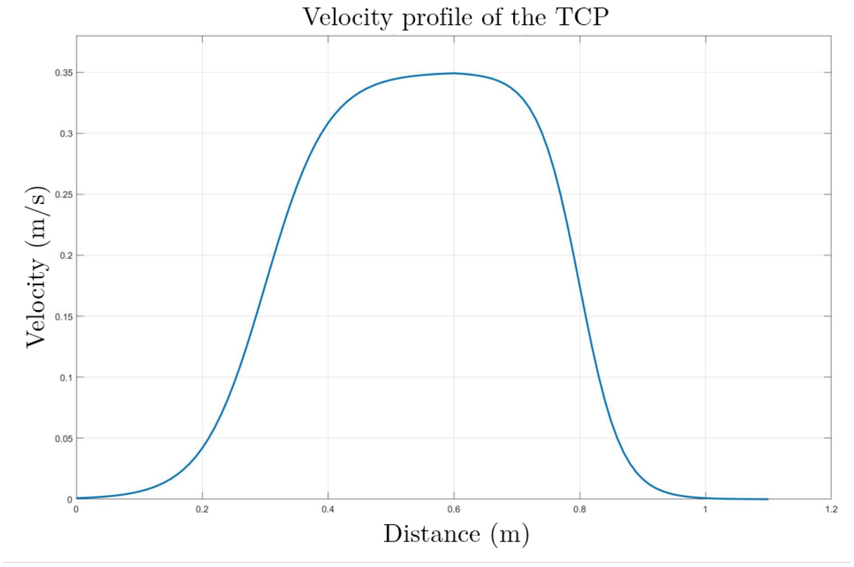


Fig. 4.10 Velocity profile of the TCP with respect to the distance from the target.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1} \left[\boldsymbol{v}_{TCP}\ \boldsymbol{\omega}_{TCP}\right]^{-T} \tag{4.16}$$

in which $\dot{\boldsymbol{q}}$ is the joint velocity vector of the robot, command to send to the robot, $\boldsymbol{J}$ is the Geometric Jacobian matrix, $\boldsymbol{v}_{TCP}$ is the vector of the linear velocities of the TCP and $\boldsymbol{\omega}_{TCP}$ is the vector of the angular velocities of the TCP.

4.3.4 Experimental tests

Experimental tests followed. The experimental setup has been previously detailed. To summarize, the camera captures the operator's gaze direction, which is processed using OpenFace 2.0. Simultaneously, the robot's position is obtained from the control unit. Then, a velocity control command is sent to the robot, that tracks the target,

i.e., the gaze focal point.
It is important to highlight that the robot's motion was controlled exclusively within the xy plane. Therefore, the focal point of the gaze was determined by projecting it onto the xy plane, at a fixed z coordinate value. At the beginning of the tests, the robot is in its home position, having joint values $q = [-180^o, -70^o, -100^o, -90^o, 90^o, 0^o]$, figure 4.11. As the focal point of the gaze gets close the UR5 workspace, the robot starts moving and aligning the TCP with the target location. The robot ceases its movement once the distance between the TCP and the gaze point reaches 40 mm. If the gaze point exits the UR5 workspace, the robot goes back to its home position.

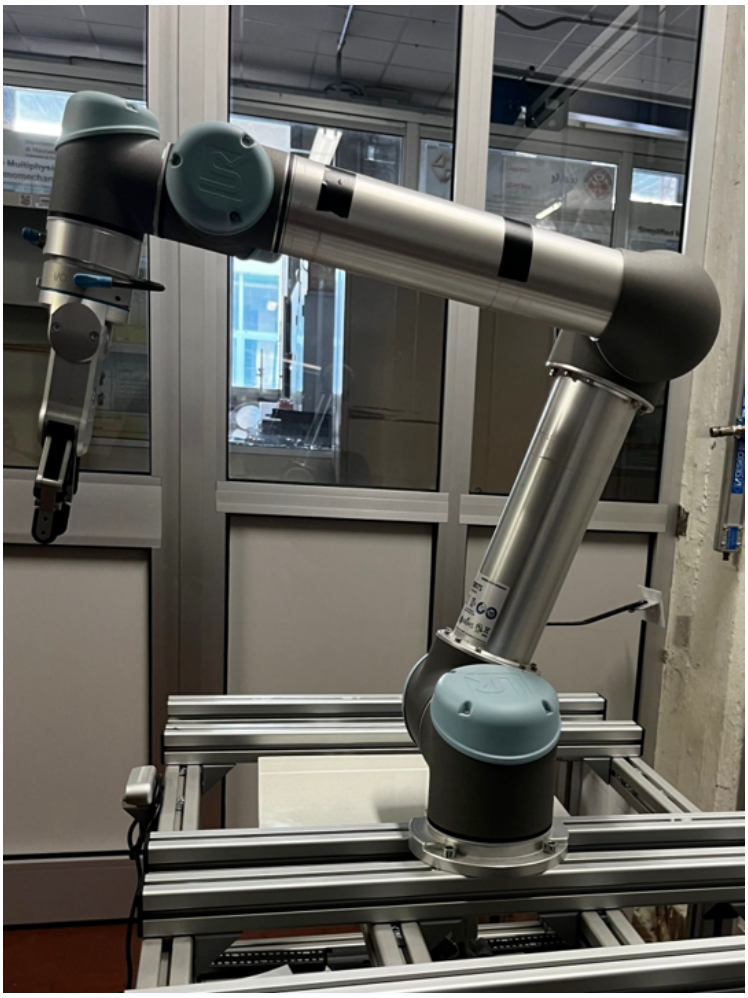


Fig. 4.11 Home configuration of the UR5 robot in gaze-tracking experimental tests.

Three different types of tests were carried out during the experiments. The first test used the unprocessed gaze signal directly. For the second and third tests, a "moving mean filter" was applied to the gaze signal to smooth out noise and random fluctuations. This filter works by taking the average of a set number of consecutive data points within a moving window, and using this average as the new, filtered value. This helps to reduce variability and produce a more stable signal. During the tests, the intention was to use the gaze to outline a rectangle.

For the first set of tests, unfiltered gaze signal is used. By examining the trajectories of both the gaze and the robot, figure 4.12, it is evident that the robot successfully followed the gaze point until the point became stationary. Then, the robot progressively approached the gaze point until it reached the predefined stopping distance. The trajectory also revealed the presence of noise in the gaze signal. In the figure, the starting position of the robot TCP and of the gaze are highlighted in a blue and a red circle, respectively. In figure 4.13 the x coordinate and the y coordinate are plot with respect to time. These plots highlight quite noise, particularly in the $y - axis$, and a delay of around 1 s. Observing 4.13, it results that segments where x remained constant corresponded to segments where y varied, and vice versa. For example, between 95.7 seconds and 116.7 seconds, the x coordinate remained approximately constant while the y coordinate showed variation.

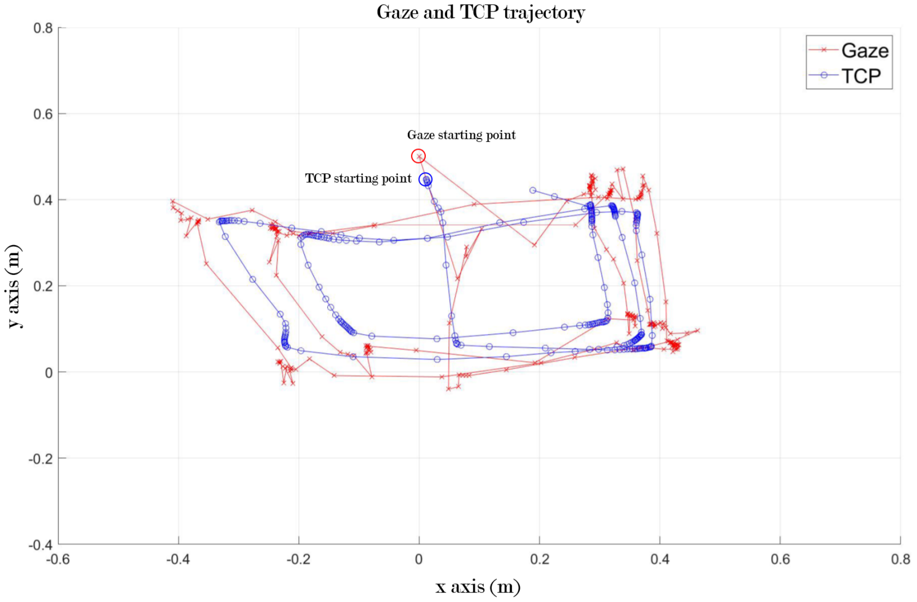


Fig. 4.12 Trajectory of gaze, in red, and TCP, in blue, for test 1, with unfiltered gaze signal.

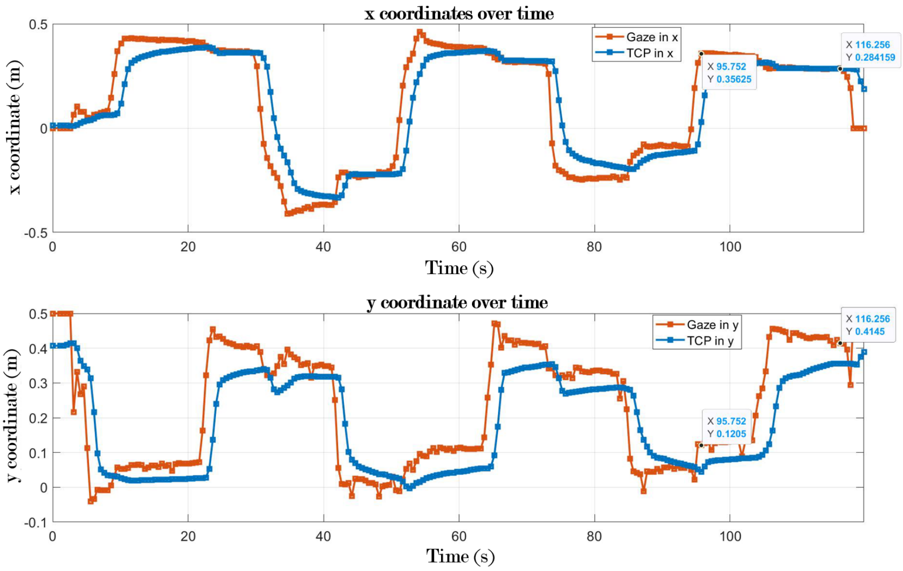


Fig. 4.13 x and y coordinates over time, for test 1, with unfiltered gaze signal.

The second tests were performed using filtered gaze signals. A moving mean filter with a window size of 5 samples was used. In figure 4.14, the trajectories of the filtered gaze data and of the robot TCP are reported. Following this, the

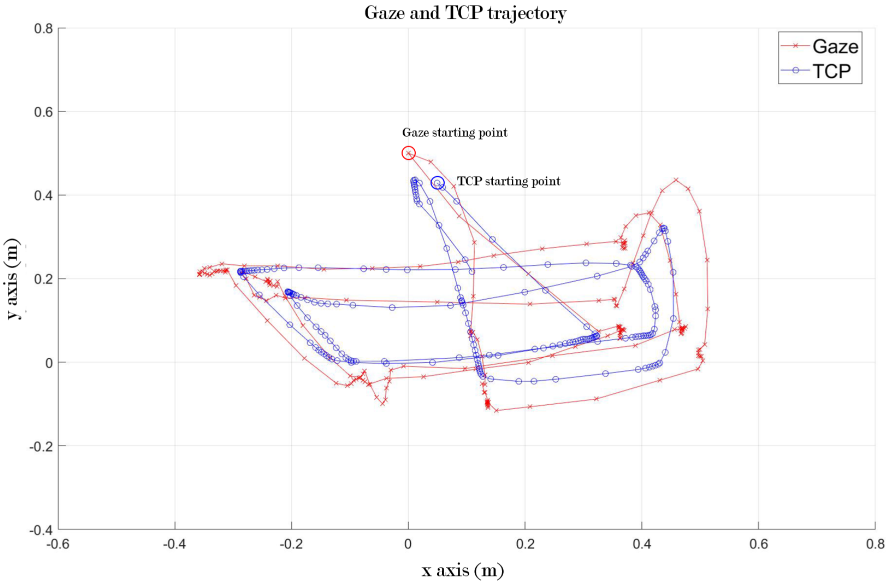


Fig. 4.14 Trajectory of gaze, in red, and TCP, in blue, for test 2, with filtered gaze signal, window size of 5.

robot gradually moved closer to the gaze point, stopping at the predefined distance. This test showed a significant reduction in noise in the gaze signal compared to the initial unfiltered test, that appears less sharp. A closer inspection of the time-based trajectories for the x and y coordinates, shown in figure 4.15, revealed much less noise, with the x-axis showing the most improvement. When analyzing the rectangle's path, a distinct pattern was observed: segments where x remained constant corresponded to segments where y varied, and vice versa. Additionally, the filtering introduced a time delay of about 3 seconds between the gaze signal and the TCP response.

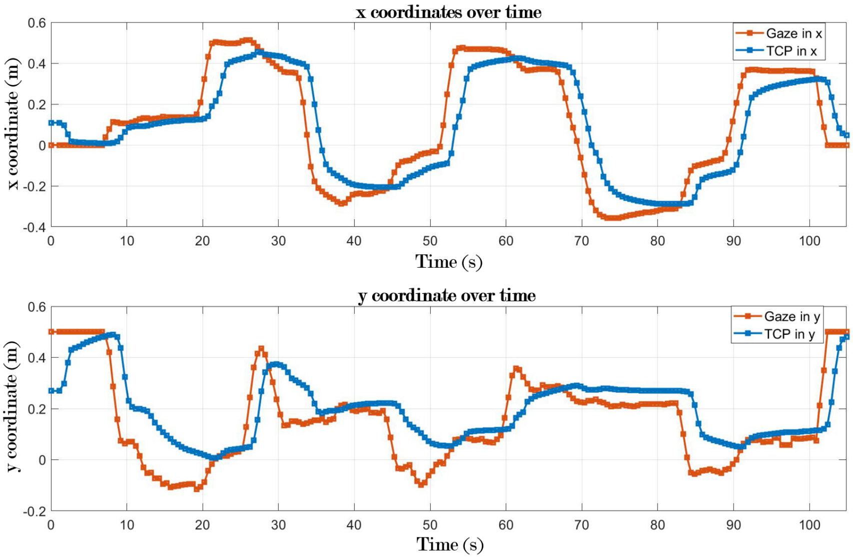


Fig. 4.15 x and y coordinates over time, for test 2, with filtered gaze signal, window size of 5.

In test 3, the gaze signals were further processed using a moving mean filter with an expanded window size of 10 samples. Analyzing the trajectories on the *xy* plane, figure 4.16, it was evident that the robot accurately followed the gaze path. Additionally, the *xy* trajectory graph indicated a reduction in noise in the gaze signal. A closer look at the temporal graphs for the *x* and *y* coordinates, shown in figure 4.17, revealed a delay of approximately 4 seconds due to the larger filter size. Additionally, the filtering caused the robot to occasionally miss specific points, particularly on the *y*-axis, as at 29 seconds, and move directly to subsequent points instead.

Among the three kind of tests conducted, the filter with a window size of 10 proved

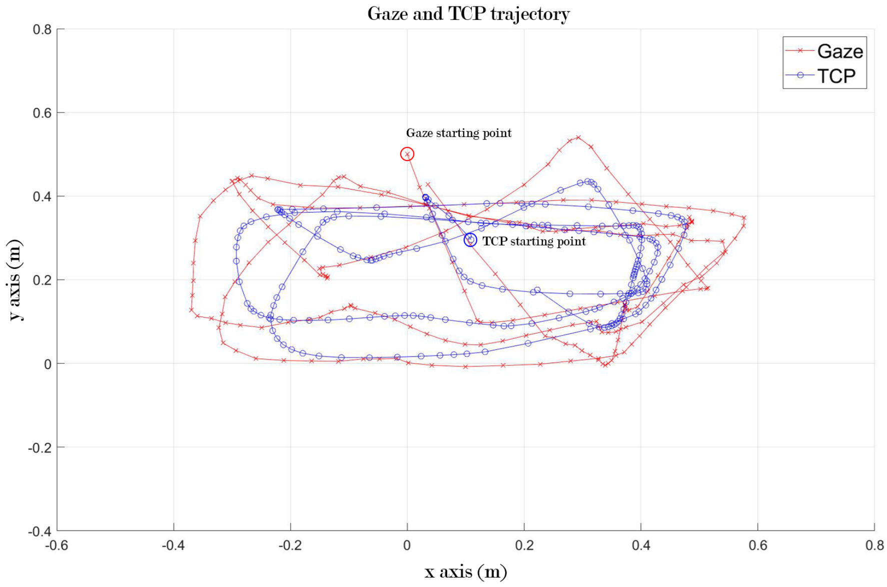


Fig. 4.16 Trajectory of gaze, in red, and TCP, in blue, for test 3, with filtered gaze signal, window size of 10.

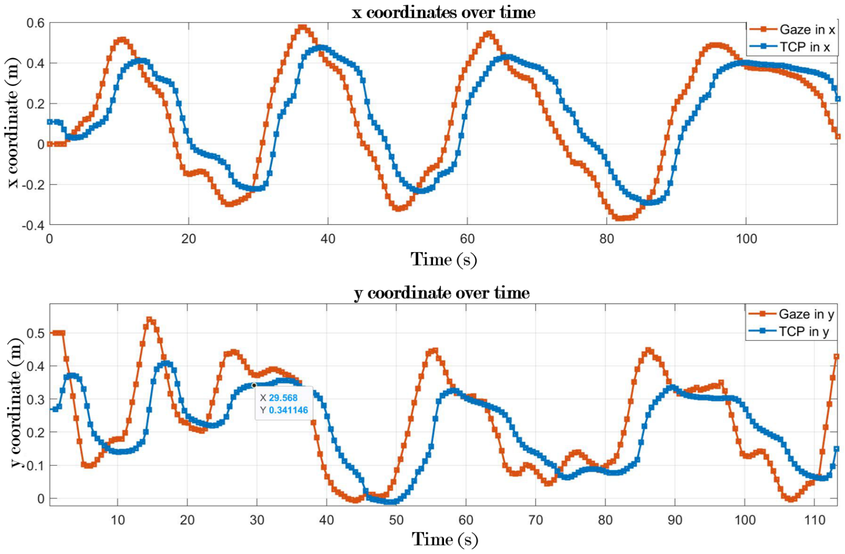


Fig. 4.17 x and y coordinates over time, for test 3, with filtered gaze signal, window size of 10.

to be the least favorable option. Although it effectively reduced noise, resulting in a smoother signal, it also introduced a substantial delay, causing the robot to deviate from precise gaze tracking. Given the need for real-time application, using the original gaze signal or the filter with a window of 5 can be both a good option. A detailed analysis of the graphs showed that the TCP tracked the gaze more accurately in the unfiltered scenario compared to the other two, while the moving mean filter with a window of 5 allows to soften uncontrolled and undesired sudden motion of the gaze, but introducing a delay.

4.4 Collision avoidance validation: mobile robotics, 2D application

Building upon the collaborative robotics algorithm introduced in section 4.2, this section explores its application to mobile robotics using a Turtlebot3 Burger robot, moving in a structured environment. The algorithm aims to condition trajectory planning for predictable paths that align with user preferences. The results obtained are presented both in a simulated environment and with a real robot, demonstrating the effectiveness of the algorithm in coordinating and optimizing the activities in a collaborative manner.

4.4.1 Collision avoidance with Turtlebot mobile robot

The Turtlebot3 Burger is a widely-used modular mobile robotics platform for research, education, and product development [123]. Its combination of features makes it an ideal choice for exploring the potential of mobile robotics. A picture of the robot is shown in figure 4.18. The Turtlebot3 Burger differential drive wheeled base provides smooth and maneuverable navigation in various environments. Its comprehensive sensor suite, including a 360^o LiDAR laser sensor, wheel encoders, and an inertial measurement unit (IMU), enables mapping, localization, and orientation. Powered by a Raspberry Pi development board, it offers the computational power and flexibility to run complex robotic software. Additionally, support for Robot Operating System (ROS) facilitates programming and application development using a widely-adopted framework. This aspect, in particular, made the robot a very good solution for algorithm testing.

In table 4.4 the characteristics of the Turtlebot3 Burger, that will be named simply

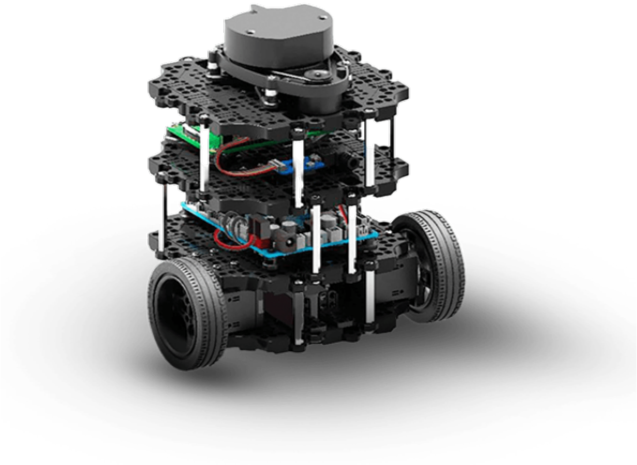


Fig. 4.18 Turtlebot3 burger.

turtlebot in the rest of the text, are reported.

Maximum translational velocity	0.22 m/s
Maximum rotational velocity	2.84 rad/s (162.72 deg/s)
Maximum payload	15 kg
Size (L x W x H)	138 mm x 178 mm x 192 mm
Weight (+ SBC + Battery + Sensors)	1 kg
Threshold of climbing	10 mm or lower
LDS(Laser Distance Sensor)	360 Laser Distance Sensor LDS-01

Table 4.4 Turtlebot3 Burger characteristics.

The gradient tracking method is employed in this work to leverage the potential field generated with local attractors. This established technique offers several advantages. Firstly, its simplicity and effectiveness ensure precise tracking of the gradient lines. Secondly, as demonstrated in [104], it is applicable to smooth artificial vector fields. The core principle of this method lies in controlling the velocity vector, rather than the acceleration vector. Indeed, turtlebot motion is controlled by commands that define linear velocity v_{tb} and angular velocity ω_{tb}, by ROS messages.
The control strategy utilizes a proportional relationship between the robot angular velocity command $\omega_{tb,set}$ and the angular error φ_{tb}:

$$\omega_{tb,set} = K \angle v_d \, v_{tb} = K \, \varphi_{tb} \tag{4.17}$$

in which K is a proportional gain, and φ_{tb} is calculated as the difference between the actual direction of the robot and the desired direction, indicated as v_d. A crucial element within this strategy is the desired direction vector v_d, which aligns with the negative gradient ∇U_t of a potential field:

$$v_d = -\nabla U_t \tag{4.18}$$

In this way, the robot is guided along the path of steepest slope in the potential field, effectively steering it towards the goal location. In figure 4.19, for a better understanding, the turtlebot variables involved in the explanation are reported.
The linear velocity command $v_{tb,set}$ hinges on the robot initial and desired positions, when the robot starts and reaches its target position with zero velocity [104]:

$$v_{tb,set} = \min\left(a_0 t, v_0, (2 a_0 d_r(t))^{\frac{1}{2}}\right) \tag{4.19}$$

Here, a_0 is the maximum acceleration allowed by the robot, v_0 is its maximum velocity and $d_r(t)$ is the position error in time, $d_r(t) = \|\boldsymbol{x}_g - \boldsymbol{x}_{tb}(t)\|$, with $x_{tb}(t)$ the actual position of the robot.

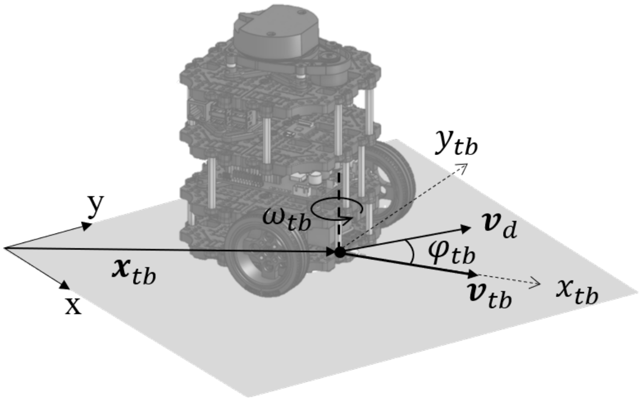


Fig. 4.19 Turtlebot3 variables involved in the gradient tracking.

4.4.2 Simulations on mobile robot

To evaluate the performance of the proposed control algorithm for guiding the robot towards its destination, a series of simulations have been conducted in a virtual

environment. The simulations allowed to assess the effectiveness of the algorithm in navigating the environment and reaching the desired destination, ensuring the obstacle is navigated through preferred area.
The simulations were performed in Gazebo, a widely used open-source 2D/3D robotics simulator. It integrates with physics engines to provide realistic robot dynamics and sensor data, as cameras and LiDARs, for perception. Additionally, Gazebo offers programmatic control interfaces and 3D visualization for testing and refining control algorithms in a safe, virtual environment. Moreover, Gazebo allows to control robots through programmatic interfaces: Gazebo programmatic interface connects seamlessly with ROS, allowing to send control commands from ROS code to the simulated robot.
The ROS Toolbox has been used to perform communication between Matlab and Gazebo, and to send velocity command to the robot, at a control frequency of 30 Hz. The velocity messages, of $/Twist$ type, are sent to the topic $/cmd\_vel$ of Turtlebot3 embedded Package.
The evaluation methodology employs a series of test scenarios, each featuring a cylindrical obstacle with a radius of R_c of 0.135 m positioned centrally between the initial and desired robot locations. The position of the obstacle is known. The tests have been performed using Matlab, in which the gradient of the potential field is performed. Referring to figure 4.20, where the radius of the external circumference of the robot is reported, it comes out that the obstacle potential field must be inflated of the dimension of the robot, so to consider the latter as a material point. So, it

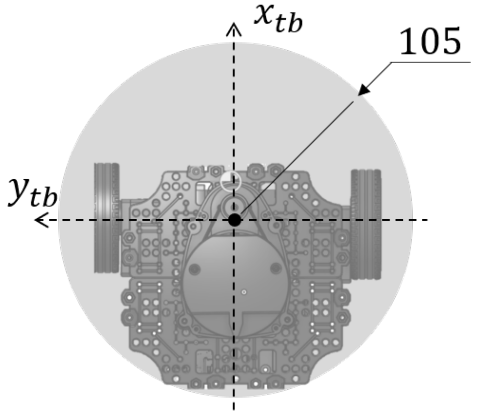


Fig. 4.20 Turtlebot3 dimension.

results R_o = (0.135 + 0.105 + 0.01) m, where 0.01 m represents a safety margin. The value of γ_o has been chosen so that the repulsive gradient is the 30% of its maximum value in correspondence of R_o, i.e., $|\nabla U_o|_{R_o} = 30\%|\nabla U_o|_{max}$. So, $\gamma_o = 80.35$. The value of β_o has been set equal to 1. The resulting potential field is shown in figure 4.21, where the robot is represented in one of its initial positions. In the first set of tests, the robot initial position $\boldsymbol{x}_i = [x_i, y_i]^T$, with $x_i = 0\,m$ and y_i between 0.05 and 0.15 m. To affect robot path without considering its approach direction towards the obstacle, a local attractor is positioned on one side of the obstacle at coordinates $\boldsymbol{x}_a = [0.64, -0.46]^T m$. The global attractor is defined at $\boldsymbol{x}_d = [2.5, 0]^T\ m$ and a with $\sigma = 0.5$. The local attractor potential field is characterized by $\alpha_a = 0.266$ and $\gamma_a = 17.02$. The strength of the local attractor is set as $\alpha_a = 0.8\tilde{\alpha}_a$, where $\tilde{\alpha}_a$ is derived from 4.13, with $x'_a = \|\boldsymbol{x}_a - \boldsymbol{x}_g\|$. The robot works with a maximum acceleration $a_0 = 0.15\ m/s^2$, a maximum velocity $v_0 = 0.1\ m/s$ and gain $K = 5$. The robot trajectory is obtained through odometry feedback. In figure 4.22 frames of test 1, with $x_i = 0.05$, inside Gazebo simulation environment are shown. The paths travelled are shown in figure 4.23(a). The robot is attracted by $\boldsymbol{x}_g$ and avoids the obstacle passing on the side of the local attractor. Its effect can be seen in figure 4.23(b), where the gradient lines of U_t are depicted. To assess the effectiveness of the local attractor in the potential field, test 2 is conducted without the local attractor, using $\boldsymbol{x}_i = [0, 0.05]^T\ m$. The outcomes are illustrated in figures 4.24(a) and 4.24(b). Unlike Test 1, the robot follows the gradient and navigates around the obstacle from the opposite direction. Additionally, due to the increased curvature of the gradient lines near the obstacle, the Turtlebot exhibits less smooth maneuvering. The final test, test 3, is conducted to examine the impact of the local attractor intensity.

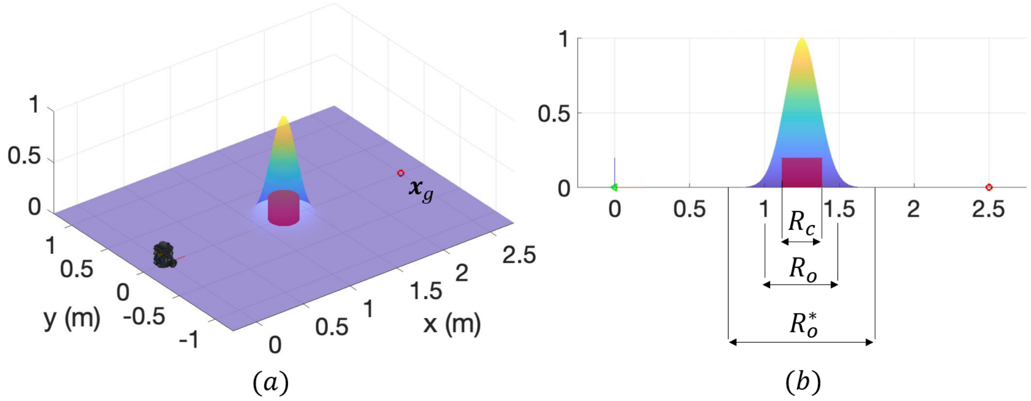


Fig. 4.21 Obstacle repulsive potential field, (a) with the robot in its initial position, and (b) in lateral view.

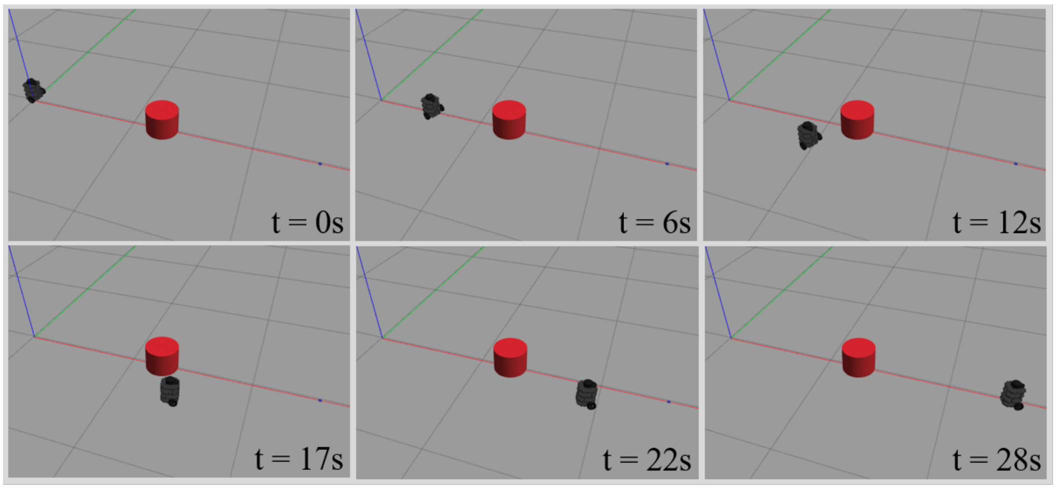


Fig. 4.22 Frames of Turtlebot simulation in Gazebo.

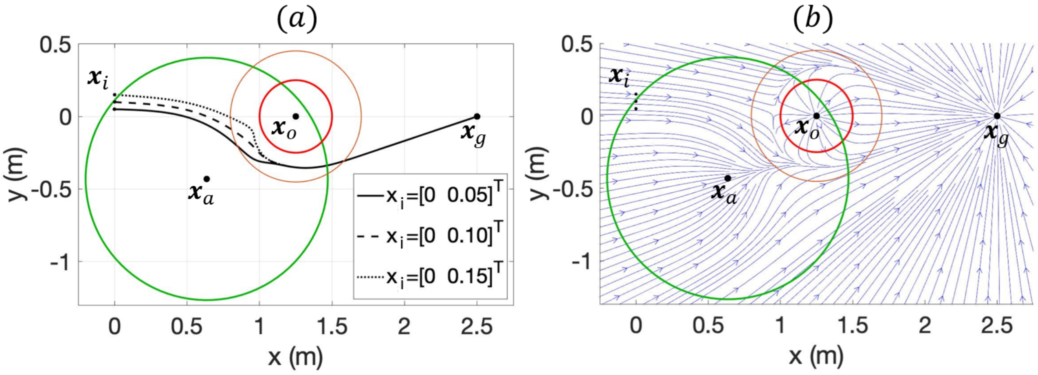


Fig. 4.23 Results of test 1; (a) paths of the robot and (b) gradient lines of the artificial potential field.

Figure 4.25(a) presents the various paths obtained from the same starting position $\boldsymbol{x}_i = [0, 0.05]^T$ m, using different values of α_a. For $\alpha_a = 0.9\tilde{\alpha}_a$, the path exhibits a sharper curve near the point $\tilde{x}$, where the saddle point would form for $\alpha_a = \tilde{\alpha}_a$, figure 4.25(b). This scenario is not ideal, as regions with high curvature result in abrupt direction changes and may overwhelm the control system. Nevertheless, with lower values of α_a, the robot can still be directed towards $\boldsymbol{x}_g$, as indicated by the dashed and dotted paths in figure 4.25(a). The tests performed in simulation environment show the efficacy and the effect of introducing the local attractor in artificial potential field, to deviate and influence the path of the robot. In next section, a real case scenario is shown.

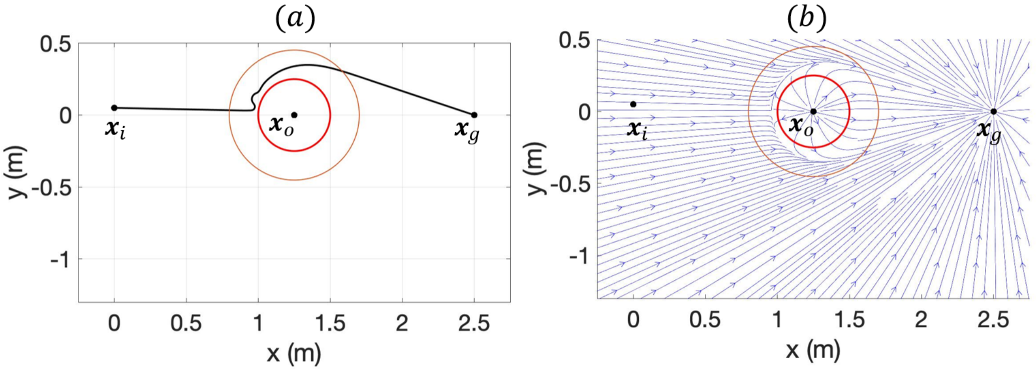


Fig. 4.24 Result of test 2: (a) path of the robot and (b) gradient lines of the artificial potential field.

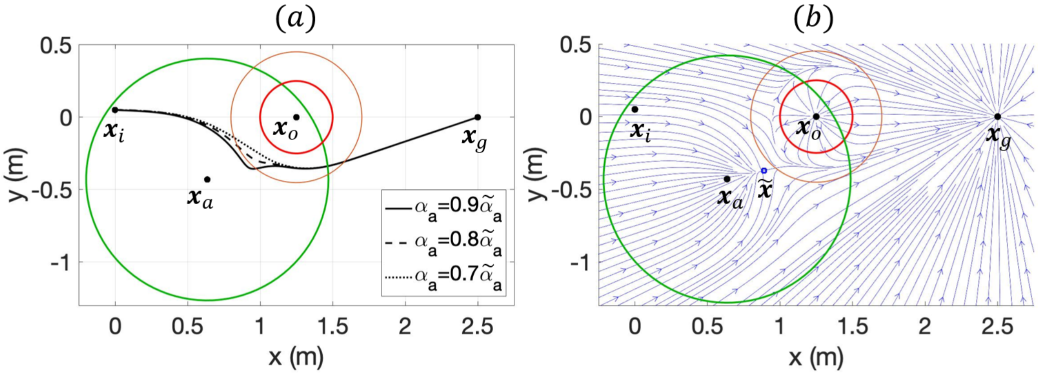
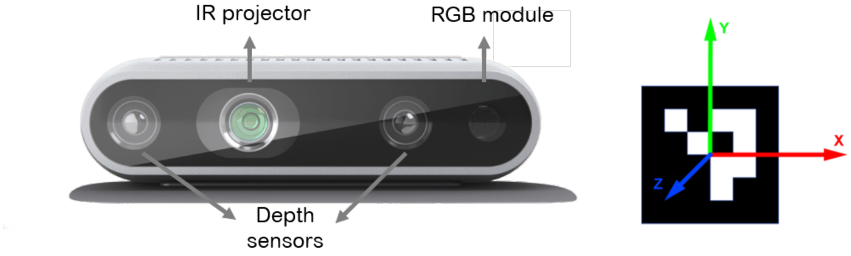


Fig. 4.25 Results of test 3: (a) paths of the robot and (b) gradient lines of the artificial potential field.

4.4.3 Experimetal tests on mobile robot using external camera for localization

Experiments were conducted in the real world using a Turtlebot3 Burger, a Realsense D435 camera [124], and Aruco markers for position feedback [125] (figure 4.26). Aruco markers are widely used in computer vision applications due to their easily detectable pose by a camera. These markers are square-shaped with a black border containing a binary matrix that encodes their unique ID number. The Aruco library, available in OpenCV, enables the reading of the marker's pose within the environment. Figure 4.26(b) illustrates an example of an Aruco marker along with its reference frame. Various dictionaries, which are collections of markers with common characteristics, exist. Aruco markers are utilized to determine the poses of the Turtlebot, obstacle, and target. The markers are detected and identified in the images captured by the Realsense camera using the OpenCV Aruco library.



(a)

Fig. 4.26 Instruments for real world tests: (a) Realsense D435 and (b) example of Aruco marker.

Experimental setup

Figure 4.27 illustrates the laboratory setup. The obstacle used is a box with an Aruco marker attached to it, designated as the $O-XYZ$ obstacle frame. A red arrow affixed to the obstacle helps to immediately identify the Y-axis of the obstacle. The attractor is virtual and defined by software, positioned along the Y-axis at a configurable distance chosen prior to the tests. Once the position of the local attractor is set, it remains fixed relative to the obstacle Aruco frame. Thus, even if the obstacle is moved during the setup, the Y-axis consistently indicates the side of the obstacle that the robot will use for collision avoidance, determining the path in advance. The workspace for the tests is determined by the camera field of view. The camera is mounted on a tripod at a height of 2 m, covering a working area of 2.5 by 1 m. The distance between the robot and the target is approximately 2 m, with the obstacle positioned in between. Multiple tests were conducted to determine the appropriate size and dictionary of the Aruco markers to use. The selected dictionary is DICT_6×6_100, which includes 100 Aruco markers. This dictionary features markers with a 6 by 6 internal matrix, each identified by an ID number ranging from zero to 99. The length of the side of the Aruco is chosen equal to 0.067 m. Under these conditions, it was measured the repeatability of the position measurement, as it was not possible to measure the exact position of the marker. Detection data is not stable, so it was collected a continuous set pictures of the Aruco, till a normal distribution of the position was obtained. The distribution was defined according to

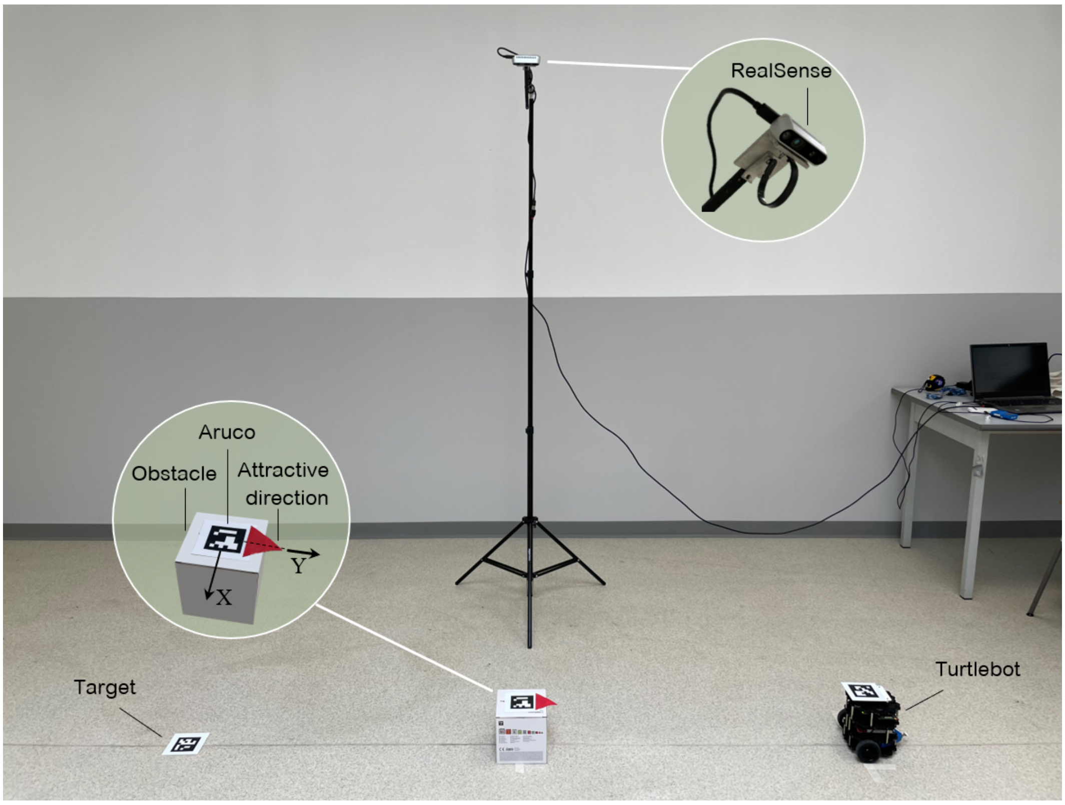


Fig. 4.27 Experimental setup, in which the turtlebot, the obstacle and the target position are shown.

the norm of the position vector. The repeatability resulted to be 0.01 m.

The Turtlebot was programmed using ROS Noetic Ninjemys on Ubuntu 20.04, and the detection of Aruco markers was handled by the OpenCV 4.2.0 library. The goal of the task is to guide the Turtlebot from its initial position to the target while circumventing the obstacle on the side where the local attractor is located. During the experiments, both the target and the obstacle remain stationary, with their positions recorded at the outset. Different IDs were assigned to the Aruco markers used for identifying the positions of the robot, obstacle, and target.

The algorithm outputs a velocity command for the Turtlebot, which is communicated via the ROS topic $/cmd\_vel$. A Python script manages the input from the Realsense camera, which includes the pose of the Turtlebot, obstacle, and target. The script calculates the potential field, determines the reference velocity, and sends this command to the robot. The primary code is tasked with computing the attractive field and generating the appropriate velocity command. Concurrently, a separate thread reads Aruco data from the Realsense camera and makes it available to the main script. A diagram illustrating the code structure is shown in figure 4.28. The Camera Thread, depicted in pink, is responsible for detecting Aruco markers and determining the Turtlebot pose. This data is written to a file, which the main thread, shown in orange, can access. To ensure atomic access to the file, a lock is shared between the two threads. The local attractor position is calculated based on the obstacle pose. Using the positions of the obstacle, the local attractor, and the target, the Sculptor block routine in the main thread calculates the potential field in which the robot operates. Once the field is established, the robot control loop runs until the position error d_r of the robot is less than a threshold value d_t. The potential field gradient is evaluated at the robot position. Utilizing the Gradient Tracking Method, which requires the robot pose $\boldsymbol{\chi}_{tb}$ as input, the robot velocity values are computed. The */cmd_vel* vector is constructed, and the velocity command is sent to the Turtlebot. The section of the code dealing with the Turtlebot, shown in purple, handles the communication between the software and the robot.

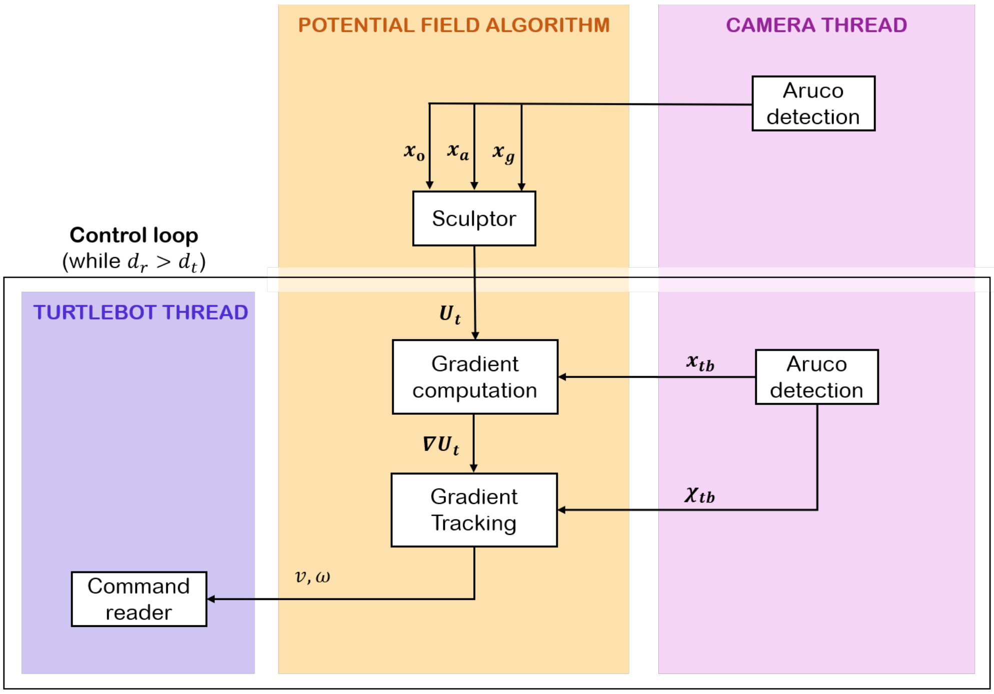


Fig. 4.28 Code schematic: the arrows indicate the flow of data, while the functions are represented within the blocks. The control loop is highlighted within the black box.

Test results

Throughout the various tests, the potential field parameters were set to the values used in previous section 4.4.2, as these proved effective during the experimental setup. Only the robot velocity and the value of α_a were adjusted to observe the robot behavior under different conditions.

The obstacle $\boldsymbol{x}_o$ is represented as a disk with a radius of $R = 0.25\ m$. Figure 4.29 illustrates the setup from the perspective of the Realsense camera. The figure also shows the robot pose $\boldsymbol{\chi}_r(t)$ at different transparencies to provide a sense of its movement. Additionally, the reference frames of each Aruco marker are depicted. The Realsense camera captures the positions $\boldsymbol{x}_s$, $\boldsymbol{x}_d$, and $\boldsymbol{x}_o$. Minor discrepancies can arise from the experimental tests since the elements are manually placed. However, the resulting positioning error is comparable to the camera position estimation error. In the subsequent figures, the robot starting position is marked by a yellow point, the target position by a magenta cross, and the path by black dots. The repulsive potential field and the obstacle are shown in red, the attractive potential field is depicted in green, and the isolines of the resulting potential field are projected in blue.

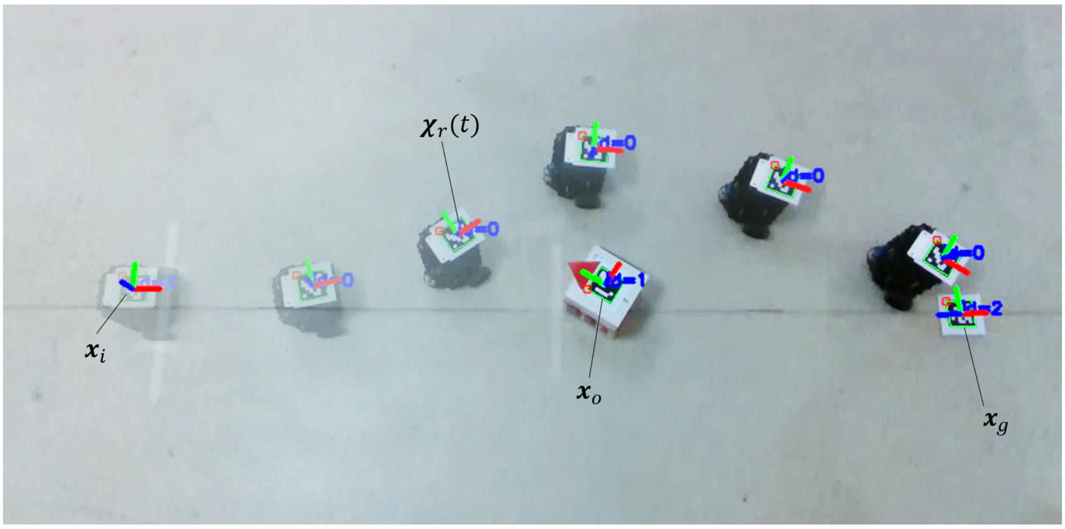


Fig. 4.29 Setup from the Realsense point of view, during an experimental test.

The first test demonstrates the robot behavior in the absence of a local attractive source, relying solely on the repulsive potential field and the global attractive potential field. According to the datasheet, the Turtlebot can reach speeds up to 0.2 m/s. For this test, a maximum velocity of $v_{max} = 0.1\ ms$ was selected. The obstacle was placed directly between the starting and target positions. Figure 4.30(a) shows the results, where the robot moves straight until it encounters the obstacle potential field. It then follows the gradient lines of the resulting potential field to reach the target. In 4.30(b), the results of test 2 are presented, showing the Turtlebot behavior when an attractor is added along the Y-axis of the obstacle reference frame at a distance of 0.75 m. The intensity of the attractive potential field is set to $\alpha_a = 0.9\tilde{\alpha}_a$. The Turtlebot turns to the left side of the obstacle, guided by the attractive source, and follows the gradient lines of the total potential field to reach the desired position. It is interesting to examine the robot behavior when the obstacle is placed on the same side as the preferred approach direction, creating a less favorable configuration. As observed in test 1, without the local attractive field, the robot would turn to the right side of the obstacle. In test 3, illustrated in figure 4.31(a), the obstacle is shifted by 0.05 m along the positive y-axis of the world reference frame, while the intensity of the attractive potential field remains at $\alpha_a = 0.9\tilde{\alpha}_a$. The robot moves smoothly, passing on the desired side of the obstacle.

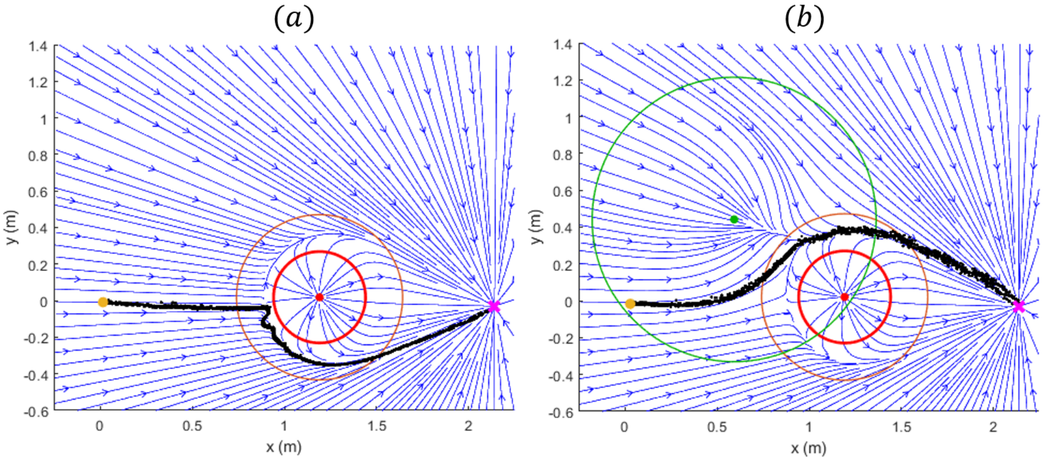


Fig. 4.30 Comparison between classical artificial potential field and the proposed method: (a) test 1: application of the classical artificial potential field without the local attractor; (b) test 2: application of the artificial potential field with the inclusion of the local attractor.

The effect of the robot velocity on its performance is also worth noting. Figure 4.31(a) presents the results of test 4, conducted under the same configuration but with a higher maximum velocity of $v_{max} = 0.2\ m/s$, represented by grey dots. At this increased speed, the robot path curves around the obstacle, where the gradient lines of the potential field change direction abruptly. The robot does not decelerate adequately, risking a collision with the obstacle.

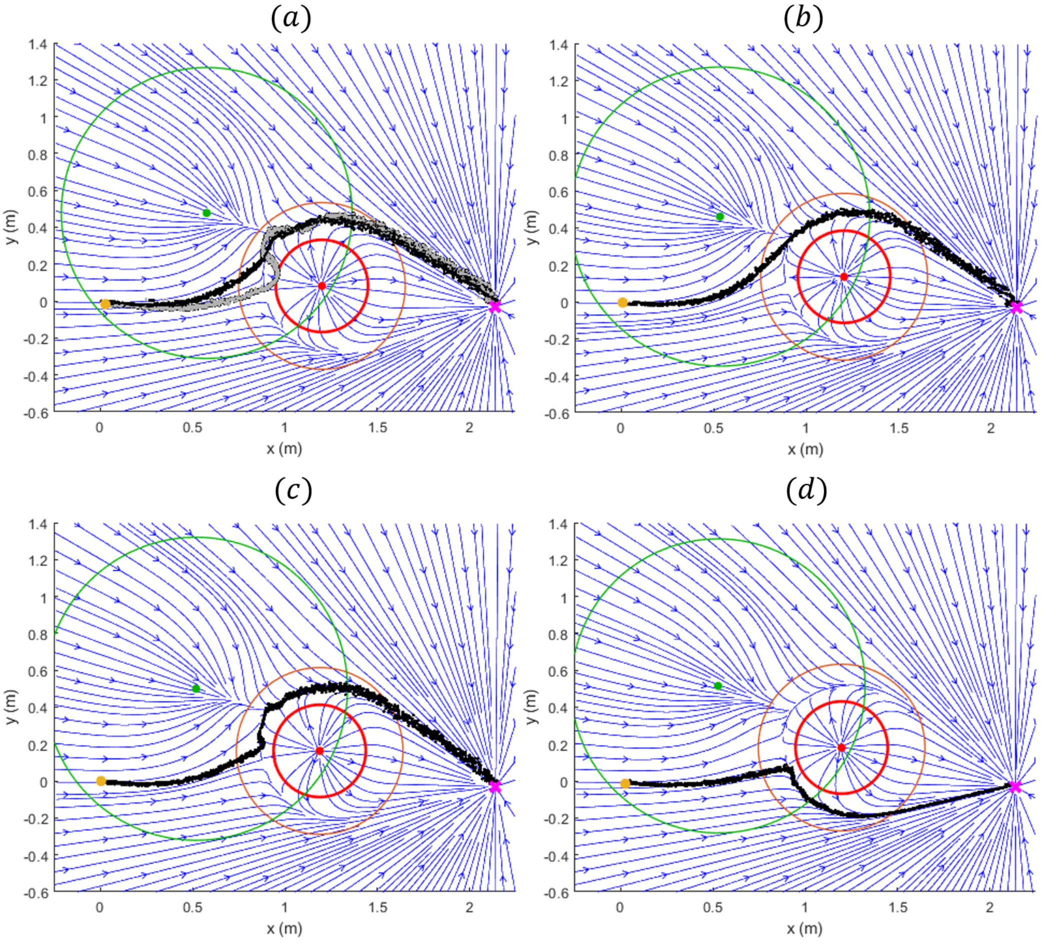


Fig. 4.31 Results of the tests performed under various conditions. (a) Comparison between test 3 and test 4. Test 3: Obstacle displaced by 0.05 m along the positive y-axis, $\alpha_a = 0.9\tilde{\alpha}_a$, $v_{max} = 0.1\ m/s$ (black dots); test 4: obstacle displaced by 0.05 m along the positive y-axis, $\alpha_a = 0.9\tilde{\alpha}_a$, $v_{max} = 0.2\ m/s$ (grey dots). (b) Test 5: obstacle displaced by 0.1 m along the positive y-axis, $\alpha_a = 0.9\tilde{\alpha}_a$, $v_{max} = 0.1\ m/s$. (c) Test 6: obstacle displaced by 0.15 m along the positive y-axis, $\alpha_a = 0.9\tilde{\alpha}_a$, $v_{max} = 0.1\ m/s$. (d) Test 7: obstacle displaced by 0.15 m along the positive y-axis, $\alpha_a = 0.7\tilde{\alpha}_a$, $v_{max} = 0.1\ m/s$.

Test 5, depicted in figure 4.31(b), illustrates the scenario when the obstacle is shifted by 0.1 m along the positive y-axis. In this test, the intensity of the attractive potential field remains at $\alpha_a = 0.9\tilde{\alpha}_a$, and the Turtlebot movement is influenced by the attractive source. Despite the obstacle being positioned on the side of the preferred path, the robot smoothly turns to the left.

An analogous test, test 6, was conducted with the obstacle moved 0.15 m in the same direction, as shown in figure 4.31(c). Here, the robot also turns to the left side of the

obstacle, but the resulting path is more swaying. This wavering path is a direct result of the shape of the potential field, with its gradient lines visible in figure 4.31(c).
To examine the effects of varying the intensity of the local attractor potential field, test 7 was performed. With α_a set to $0.7\tilde{\alpha}_a$, the resulting gradient lines under the same conditions as the previous test failed to guide the robot to turn to the left and desired side of the obstacle, as if the robot did not perceive the influence of the attractive source, as depicted in figure 4.31(d).
Experimental tests corroborate the findings obtained from the simulations in previous work. The method is applied to static obstacle, but can be applied to moving ones. In this section a validation of the theoretical formulation is presented.

4.5 Conclusions

In this chapter, the integration of gaze tracking and collision avoidance algorithms within a collaborative robotic framework has been investigated. The experimental results confirm the validity and robustness of the proposed approach. Key findings include:

1. Effective Integration of gaze tracking: The integration of gaze tracking with the UR5 robot was successfully achieved using OpenFace 2.0 and CoppeliaSim, demonstrating a seamless interaction between the operator's gaze and the robot motion. This integration underscores the potential for real-time gaze-based control in robotic applications.

2. Collision Avoidance: The developed collision avoidance algorithms proved to be effective in dynamically adjusting the robot's trajectory to prevent collisions. The use of local attractors significantly enhanced the robot ability to navigate around obstacles smoothly and efficiently, giving the possibility to deviate the path of the robot in according to user preferences.

3. Real-world Applicability: The real-world tests with the UR5 with the Astra camera and with the Turtlebot3 and the Realsense camera highlighted the practicality and effectiveness of the system in a laboratory setting.

In summary, the chapter puts in evidence that the combination of gaze tracking and collision avoidance algorithms can significantly enhance the capabilities of collaborative robotic systems. This approach not only improves the safety and efficiency of human-robot interactions but also opens up new possibilities for the application of collaborative robotics in various industrial and research settings.

Chapter 5

Collaborative robotic cell

The results and methods described in the previous chapters were integrated into a prototype of collaborative robotic cell, designed to ensure safe cooperation between human and robot. Within this collaborative cell, the aspects involved in the disassembly of an EVB are examined. The disassembly of the EVB serves as an example, illustrating how the developed logic and robotic algorithms can be applied to the assembly or disassembly of any large-scale component, that cannot be fully automated. The primary goal is to demonstrate the feasibility and effectiveness of using collaborative robotics in complex tasks that require human-robot interaction, and the effectiveness in this task of the algorithms developed in this thesis. By integrating advanced robotic systems, vision sensors, and sophisticated algorithms, the robotic cell aims to enhance safety, efficiency, and precision in the disassembly process.

The collaborative robotic cell is designed to address the challenges associated with the disassembly of large and complex components, such as EVBs, where full automation is not feasible due to variability and complexity. The results of this work can be extended to other applications, including assembly, maintenance, and repair tasks in various industrial settings. By demonstrating the practical implementation of collaborative robotics, this work aims to pave the way for more widespread adoption of such systems in industrial environments.

In this chapter, the final experimental tests in a robotic cell are shown, according to the results of chapter 2. The collision avoidance between the robot and the moving operator are shown, the strategy for the localization of the robot is explained and the collision avoidance of the robot arm with the operator during disassembly task is tested.

5.1 Setup of the final system

This section presents the final setup selected for the battery disassembly process. It details the robot utilized, the cameras chosen, and the solution devised to emulate the battery system.

The final system comprises the robotic systems, the cameras and a working table. For the selection of the collaborative robotic arm, five models have been considered. The robots are shown in table 5.1, with their characteristics. The fundamental necessity is the possibility to program the robot in ROS, as it is particularly suitable for research activity and for the integration of different components, as robots, cameras and sensors. Only the UR5 e-series and the Franka robot FR3 present this possibility. The two robots primarily differ in the higher payload of the UR5 robot, 5 kg compared to 3 kg, and the greater number of axes of the Franka robot, 7 axes compared to 6 axes. It was considered preferable for the development of research

Table 5.1 Collaborative robotic arms and their characteristics.

UR 5 e-series	**KUKA LBR iiwa**	**Franka Robot FR3**	**ABB 2-Arm YUMI Robot**	**OMRON TM5-9000**
Official ROS Library	No Official ROS Library	Official ROS Library	No Official ROS Library	No Official ROS Library
5 kg Payload	7 kg Payload	3 kg Payload	0.5 kg Payload per arm	4 kg Payload
850 mm Reach	800 mm Reach	855 mm Reach	559 mm Reach	900 mm Reach
6 dof	7 dof	7 dof	7 dof per arm	6 dof

activities to introduce a 7-axis machine, which could potentially allow for further improvements in the algorithms under development, even if it means sacrificing some payload capacity. In contrast, having a higher payload is not seen as significantly beneficial for the development of the robotic cell.

A similar analysis has been done for the selection of the mobile robot. The four considered robots are shown, with their characteristics in table 5.2. The MiR250 robot does not give the possibility to be programmed in ROS, it cannot be considered for the activity. The other three robots can be programmed in ROS. The Robotnik SUMMIT-XL STEEL would be ideal due to its omniwheels, but its cost is the double the cost of the other robots considered. The trade-off is made between the TIAGo platform from PAL Robotics and the Robotic Husky from Clearpath. These two platforms are substantially different in their architecture: the TIAGo platform has two drive wheels and controls motion through independent speed control of these wheels; it is primarily designed for indoor use. The Clearpath platform has four wheels, two of which are steering wheels, and is also designed for outdoor environments. Since the prototype layout is expected to be implemented in an indoor environment and the solution with two independent wheels is better suited for use in confined spaces, the PAL Robotics platform was selected.

Table 5.2 Mobile robots and their characteristics.

MiR 250	**Clearpath Robotics Husky**	**PAL ROBOTICS TIAGo Base**	**Robotnik SUMMIT-XL STEEL**
No Official ROS Library	Official ROS Library	Official ROS Library	Official ROS Library
250 kg Payload	75 kg Payload	100 kg Payload	250 kg Payload per arm
Footprint LxWxH 990x670x390 mm	Footprint LxWxH 990x670x390 mm	Footprint ØxH 540x300 mm	Footprint 978x776x510 mm

5.1.1 Vision systems

To ensure accurate detection and tracking within the collaborative robotic cell, two types of vision systems were employed: two Microsoft Kinect V2 and an Intel Realsense D435. The Microsoft Kinect V2 sensors are used to track the human operator movements by capturing their skeleton, facilitating collision avoidance. On the other hand, the Intel Realsense D435 is mounted on the robotic system to enhance spatial synchronization, allowing precise interaction with the workbench.

Microsoft Kinect V2

The Microsoft Kinect V2 sensor [126] is utilized in the collaborative robotic cell for its advanced depth-sensing capabilities and wide field of view. This sensor, figure 5.1 provides detailed depth information, which is crucial for accurately detecting and tracking the position of the human operator. The Kinect V2, in particular, includes in its SDK the possibility to track directly the skeleton of the user. This is essential for collision avoidance and ensuring the safety of the human operator during the disassembly process. Throughout this text, the Microsoft Kinect V2 will be referred to simply as Kinect.
The technology for depth-sensing is the time-of-flight (TOF). In table 5.3, the technical specification of the Kinect are reported. The Kinect V2 can be programmed in MATLAB, facilitating easy integration with the robotic system for data analysis and algorithm development.

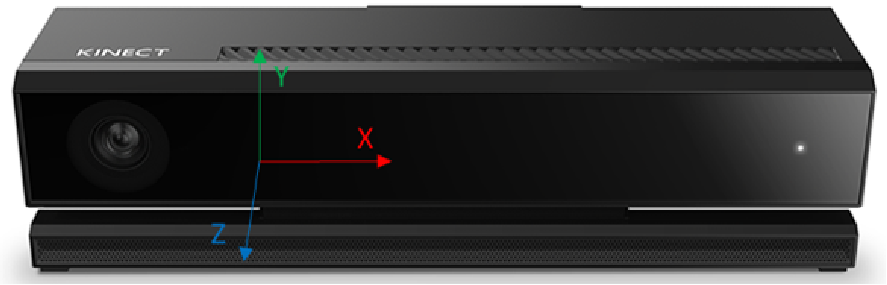


Fig. 5.1 Picture of Microsoft Kinect V2, with its reference frame.

The problem of using a single camera is that occlusion can take place, and some parts of the skeleton of the operator can be lost for a certain amount of time. So, two Kinects are used. The Kinect sensors are placed in positions that maximize their

Table 5.3 Specifications of the Microsoft Kinect V2

Feature	Specification
Resolution (Color)	1920 x 1080 pixels
Resolution (Depth)	512 x 424 pixels
Field of View	70 x 60 degrees
Range	0.5 to 4.5 meters
Frame Rate	30 frames per second
Infrared (IR) Capabilities	Yes
SDK Support	Skeleton tracking, gesture recognition

coverage of the working area, above the workbench and at angles that allow them to capture the full range of motion of the human operator. The skeleton tracking of the two devices are fused. The data from these sensors are processed in MATLAB, where the skeleton tracking information is combined. Spatial Matching involves transforming the 3D data from each Kinect sensor to a common reference frame, known as the world frame. This is achieved by calculating a transformation matrix that aligns the depth frames from each Kinect to the world frame. Solid markers placed within the workspace are used to facilitate this process, as shown in figure 5.2. By identifying these markers in the point clouds from each Kinect, their positions can be mapped accurately to the world frame. Three cones having spherical markers one their top are placed on the table. These markers are designed so that their geometry can be distinguished in the point cloud of the Kinect, up to a distance of 2.5 *m* from the IR sensor. The coordinates of these points can be identified with an average depth accuracy error of less than 2 *mm*. The reference frame is built on the first marker, and the skeleton data are referred to it.

Intel Realsense D435

The Intel RealSense D435 sensor, figure 4.26(a) is chosen for its high precision and compact dimensions. The RealSense D435 is mounted on the robotic system to enhance spatial synchronization, enabling the robot to accurately work on the workbench and perform precise tasks. The key specifications of the RealSense D435 are summarized in the table 5.4. The Realsense D435 data, that will be referred to simply as Realsense in the text, can be taken and manipulated using OpenCV package in Python, facilitating the integration with the robot. It is used to determine

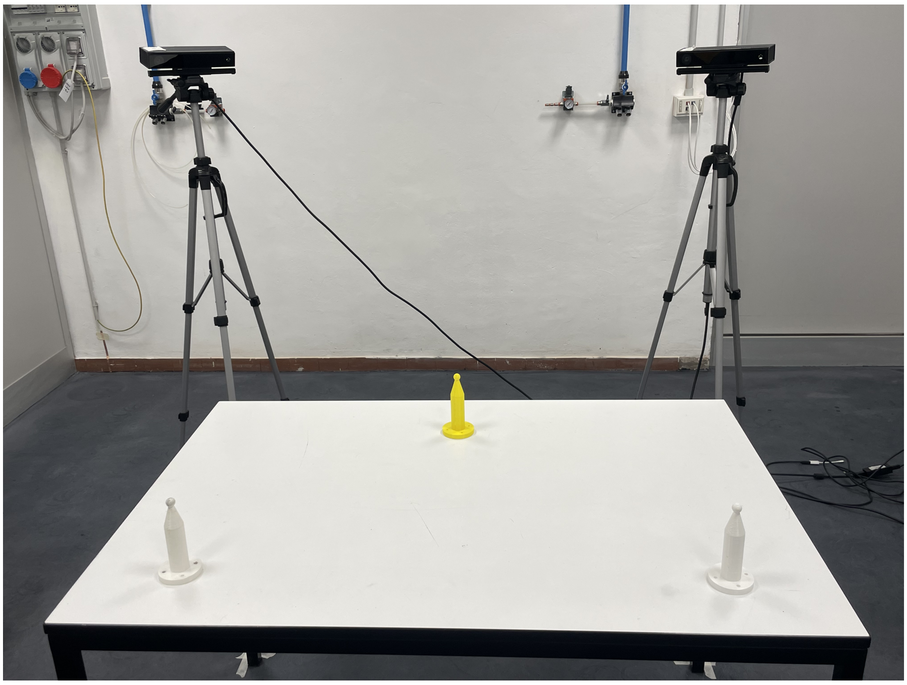


Fig. 5.2 Solid markers used for Kinect spatial matching.

Table 5.4 Specifications of the Intel RealSense D435

Feature	Specification
Resolution (Color)	1920 x 1080 pixels
Resolution (Depth)	1280 x 720 pixels
Field of View	85 x 58 degrees
Range	0.2 to 10 meters
Frame Rate	Up to 90 frames per second
Infrared (IR) Capabilities	Yes
SDK Support	Depth sensing, 3D scanning

the exact position of the robotic system, particularly the Franka FR3 robotic arm, relative to the workbench. This is achieved by reading the poses of Aruco markers placed within the workspace. By recognizing these markers and calculating their pose, the Realsense can accurately determine the spatial location of the robotic arm.

5.1.2 Description of the robotic system

In this section, the key components of the collaborative robotic system are described: the Tiago Base mobile robot from PAL Robotics and the Franka Emika robotic arm. Additionally, the way these two robots have been integrated to work seamlessly together within the collaborative cell are explained. Detailed descriptions of each robot are provided below, followed by the integration methods used to combine their functionalities into a cohesive system.

TIAGo Base from PAL ROBOTICS mobile robot

The TIAGo Base from PAL Robotics, figure 5.3, has been selected as the mobile platform due to several key reasons: it features a differential drive kinematics allowing it to rotate on its own axis and navigate in tighter spaces, it offers a good compromise within the available budget. As the MiR250 AGV, the TIAGo base has 2 driving wheels and 4 caster wheels, making the dynamic model of the robot adaptable to the TIAGo base, too. Additionally, the Tiago Base can be easily programmed using ROS. For brevity, the TIAGo Base will be referred to simply as Tiago throughout this document. Tiago is designed to support a variety of applications, including those requiring high mobility and precision. It uses Simultaneous Localization and Mapping (SLAM) to navigate autonomously within the workspace, avoiding obstacles. Equipped with a laser range finder, Tiago can perceive its environment, aiding in navigation and task execution. The key specifications of the TIAGo Base are summarized in the table 5.5, [127].

Table 5.5 Specifications of the TIAGo Base

Feature	Specification
Payload Capacity	50 kg
Navigation	SLAM-based autonomous navigation
Wheels	2 x motorised and 4 x caster
Speed	Up to 1 m/s
Height	300 mm
Footprint	∅ 540 mm
Weight	40 kg
Laser	Hokuyo URG-04LX-UG01 up to 5.5 m
Control Interface	ROS-based

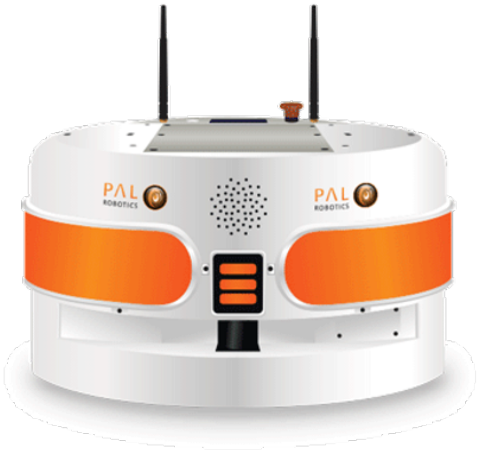


Fig. 5.3 Picture of the TIAGo Base.

Franka Robot FR3

The Franka Robot FR3, figure 5.4, has been selected for its precision and versatility. This robotic arm, typically offering 7 degrees of freedom, is treated as a 6-degree-of-freedom robot in this work. The decision to use it as a 6-axis robot simplifies certain aspects of integration and control within the current system. However, there is potential for future investigations into leveraging the additional degree of freedom to enhance system capabilities. Additionally, the Franka FR3 is integrated with ROS, enabling seamless communication and control within the collaborative robotic cell, and simplify its integration into the overall system, allowing for efficient task execution and control. For brevity, the Franka Robot FR3 will be referred to simply as Franka throughout this document. The key specifications of Franka are summarized in the table 5.6, [128].

Franka is also equipped with the Franka Hand [128], figure 5.5 a dexterous and adaptive gripper that enhances its ability to manipulate objects accurately and efficiently. The Franka Hand is a parallel gripper with exchangeable fingers, providing a continuous grasping force of 70 N and a maximum force of 140 N. It offers a travel distance of 80 mm, and has a weight of 0.7 kg. The DH parameters provide a systematic way to describe the geometry of robotic arms. For the Franka Emika Robot FR3, the modified DH parameters are reported in table 5.7, [129].

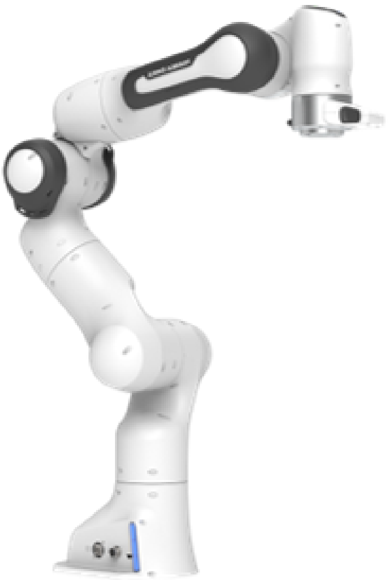


Fig. 5.4 Picture of Franka Robot FR3.

Table 5.6 Specifications of the Franka Emika Robot FR3

Feature	Specification
Payload Capacity	3 kg
Reach	855 mm
Repeatability	±0.1 mm
Degrees of Freedom	7 revolute joints
Control Interface	ROS-based
Weight	18 kg
Joint Speed	150 °/s for A1-A4 and 301 °/s for A5-A7
Cartesian velocity limit	2.0 m/s

Table 5.7 Denavit-Hartenberg Parameters for the Franka Robot FR3

Joint	θ_{DH} [rad]	a_{DH} [m]	d_{DH} [m]	α_{DH} [rad]
1	θ_1	0	0.333	0
2	θ_2	0	0	-$\pi/2$
3	θ_3	0	0.316	$\pi/2$
4	θ_4	0.0825	0	$\pi/2$
5	θ_5	-0.0825	0.384	-$\pi/2$
6	θ_6	0	0	$\pi/2$
7	θ_7	0.088	0	$\pi/2$
Flange	0	0	0.107	0

Fig. 5.5 Picture of Franka Hand.

In addition, the transformation matrix from the flange in the end effector (EE) and the center between the fingertips of the gripper is:

$$
{}^{F}\boldsymbol{A}_{TCP} = \begin{bmatrix} 0.707 & 0.707 & 0 & 0 \\ -0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0.1034 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}
$$

Integration of Franka FR3 and Tiago Base

The integration of the Franka robotic arm with the Tiago Base involves combining mechanical and electrical components to form a cohesive and functional system within the collaborative robotic cell. An aluminum profile structure from Item [130] was mounted on the Tiago Base using the designated holes on its base, figure 5.6. The bigger diagonal of the rectangle formed by the profiles on the Tiago robot is equal to $0.60m$. The controller for the Franka FR3 was mounted on the aluminum structure, ensuring that it is securely held and easily accessible for maintenance and connection. Additionally, a portable power station, a battery with a capacity of 256 Wh, that allows 2 hours of battery life, was placed above the controller to serve as the power supply for Franka. This battery provides the necessary power to operate the robotic arm and ensures continuous operation during tasks, with an output capacity of up to 600 W. To enhance the system's spatial awareness, a horizontal bracket was installed on the aluminum structure to mount the RealSense D435 camera, enhancing the robot ability to synchronize spatially with the workbench.
Both robots are controlled via dedicated computers running Ubuntu 20.04, and the

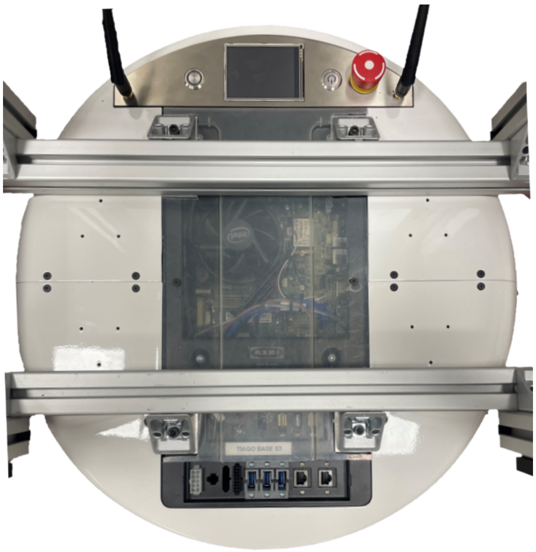


Fig. 5.6 TIAGo Base and aluminum structure fixed on it.

robots are operated using ROS.

The final robotic system is shown in figure 5.7.

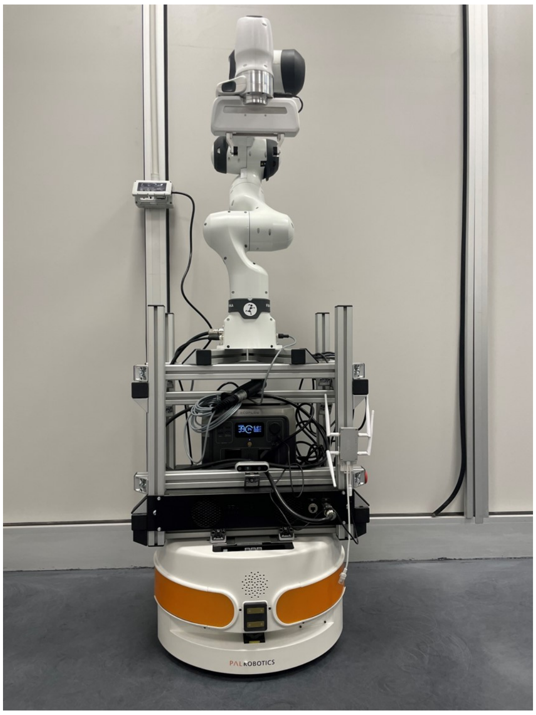


Fig. 5.7 Final arrangement of the robotic system.

5.1.3 Final robotic collaborative cell setup

In figure 5.8, the final layout of the robotic cell is sketched. The layout includes a workbench where the battery to be disassembled is placed, with some table positions identified by Aruco markers. These markers allow the robot to accurately localize itself relative to the table as it moves around it, ensuring that the robotic arm can effectively operate on the battery. By reading the positions of the Aruco markers, the system can maintain precise spatial awareness, which is crucial for accurate and efficient task execution. Two Kinect V2 cameras are positioned to track the skeleton of the human operator, ensuring safety and collision avoidance. Both the Tiago base and Franka robotic arm move within the environment, coordinated through ROS and controlled via dedicated computers.

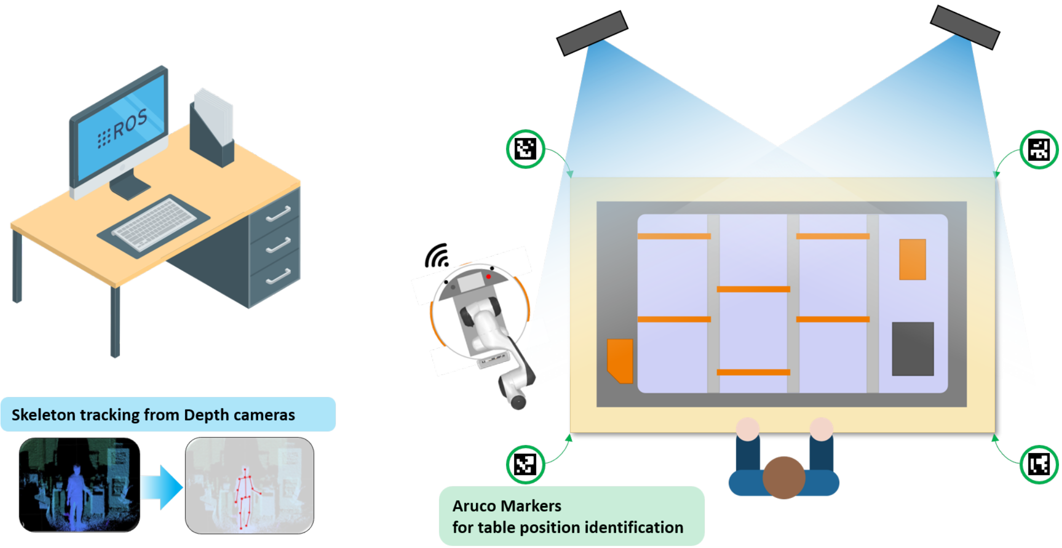

ROS
Skeleton tracking from Depth cameras
Aruco Markers
for table position identification

Fig. 5.8 Final layout of the robotic cell.

To replicate a real battery, a 3D-printed mockup was created, comprising only the initial part of an actual battery, limiting to the section of the first battery module, as can be seen in figure 5.10. Since the work focuses on just a portion of the battery, figure 5.9, only three sides of the workbench can be exploited, with the rest of the battery assumed to extend beyond the table. Therefore, the two kinect refer the tracked skeleton of the person to a fixed reference system on the table, with a known transformation matrix relative to the Aruco markers, achieving complete synchronization of the system.

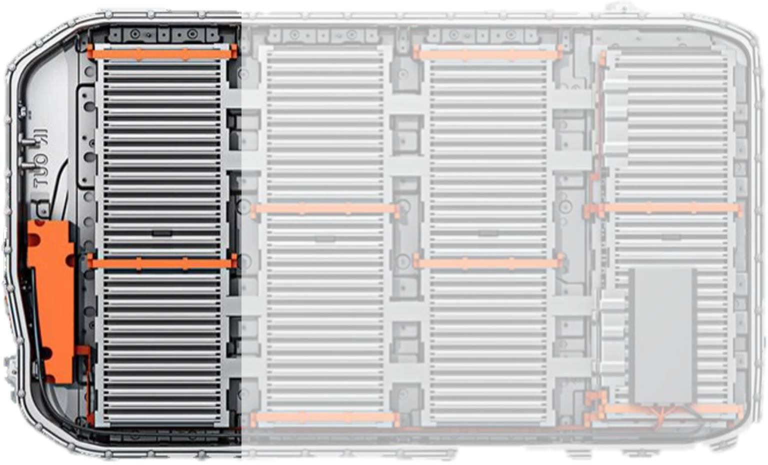


Fig. 5.9 Complete battery, shaded and full color, and printed portion, full colors.

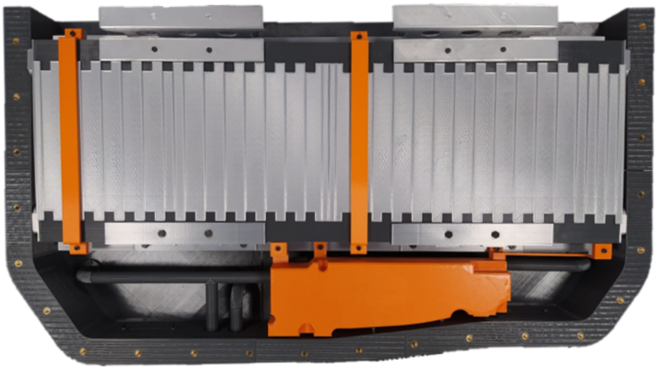


Fig. 5.10 3D-printed mockup of the battery.

5.1.4 Improvement of localization with aruco markers

Precise positioning of the robot is crucial to ensure accurate and efficient operation. The robotic system moves in the robotic cell through the Tiago Base mobile platform. However, the Tiago Base has a positioning error of approximately 50 mm, which is too large for the precise positioning required for the Franka robotic arm to perform detailed tasks on the battery.

To enhance the positioning accuracy, Aruco markers were strategically placed on the legs of the workbench at a height detectable by the Realsense camera mounted on the robotic system. All the Aruco marker have different ID number, so that this information can be used to determine the where the robot is with respect to the table. A reference frame $O_b - x_b y_b z_b$ was established directly on the battery, and the poses of the Aruco markers ${}^{b}\boldsymbol{A}_{AM}$ relative to this reference frame were carefully measured and recorded. The poses of the battery components are defined relative to this reference frame, ${}^{b}\boldsymbol{A}_{obj}$. By transforming the poses of the Aruco markers to the reference frame on the battery, it becomes possible to accurately determine the position of the battery components relative to the Arucos, ${}^{AM}\boldsymbol{A}_{obj}$, and so to the table, as:

$$ {}^{AM}\boldsymbol{A}_{obj} = \left({}^{b}\boldsymbol{A}_{AM}\right)^{-1} {}^{b}\boldsymbol{A}_{obj} \tag{5.2} $$

Furthermore, a transformation matrix between the camera and the Franka robot was calculated, ${}^{FR}\boldsymbol{A}_{RS}$. For this operation, an Aruco marker was used. The marker was attached to a 3D-printed tile with a central extruded rectangle of the same size as the marker to ensure precise placement. The tile was printed in white to ensure good readability of the marker. The marker was then positioned at the center of the Franka Hand gripper, aligning the marker center with the robot tool center point TCP. The robot then presented the marker to the Realsense camera in three different poses, recording the pose of the Aruco marker both in the robot base reference frame, i.e., reading the TCP pose, and in the Realsense reference frame using the OpenCV library. An auxiliary reference frame is constructed in one of the three points, and the transformation matrices of this reference frame relative to both the Franka base and the Realsense reference frame are constructed according to equations 3.51 - 3.56. In this way, the transformation matrix between the Franka base and the Realsense is calculated. The accuracy of the matrix calculated is 2 mm. The Realsense reads the pose of the Aruco, ${}^{RS}\boldsymbol{A}_{AM}$, and returns the pose of the Aruco and its ID, so that the correct transformation matrix ${}^{b}\boldsymbol{A}_{AM}$ is used. Therefore, the position of the object on the battery is defined with respect to the Realsense reference frame ${}^{RS}\boldsymbol{A}_{obj}$:

$$ {}^{RS}\boldsymbol{A}_{obj} = {}^{RS}\boldsymbol{A}_{AM}\,{}^{AM}\boldsymbol{A}_{obj} \tag{5.3} $$

Once known the transformation between the Franka robot and the Realsense, it is straightforward to calculate the pose of the object with respect to the robot:

$$ {}^{FR}\boldsymbol{A}_{obj} = {}^{FR}\boldsymbol{A}_{RS}\,{}^{RS}\boldsymbol{A}_{obj} \tag{5.4} $$

In figure 5.11, the worktable setup is shown. The Realsense reads the pose of the marker, which is sent to the algorithm in order to calculate the pose of the desired object on the battery with respect to the Franka robot, and allowing an accurate positioning.

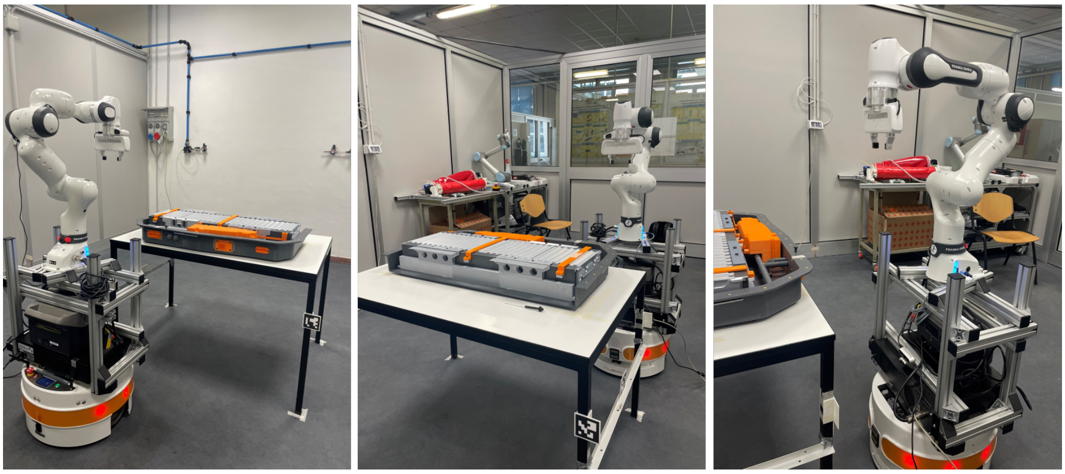


Fig. 5.11 Setup of the worktable, with the Aruco markers, and the robotic system with the Realsense.

5.2 Collision avoidance on the robotic arm

The collision avoidance between the robotic arm and the operator must be implemented, too. In this contest, the collision avoidance algorithm calculates a repulsive velocity for each robot joints to escape the operator's arms.
As anticipated, the robotic arm Franka Emika is simplified as a 6-dof robotic arm. Either the third or the last joint can be fixed, in order to obtain an equivalent 6-axis robot. Therefore, the generic pose of the EE of the robot $\boldsymbol{\chi}_{EE} = [\boldsymbol{p}_{EE}\ \boldsymbol{\varphi}_{EE}]^T$, with $\boldsymbol{p}_{EE}$ the 1×3 position vector of the EE and $\boldsymbol{\varphi}_{EE}$ the 1×3 orientation vector.
A scenario is considered where the robot needs to reach a desired EE pose $\boldsymbol{\chi}_{EE,f} = [\boldsymbol{p}_{EE,f}\ \boldsymbol{\varphi}_{EE,f}]^T$, starting from its initial pose $\boldsymbol{\chi}_{EE,s} = [\boldsymbol{p}_{EE,s}\ \boldsymbol{\varphi}_{EE,s}]^T$. In the absence of obstacles, the path would be a straight line connecting $\boldsymbol{p}_{EE,s}$ and $\boldsymbol{p}_{EE,f}$.
However, real-world environments are rarely obstacle-free. This necessitates the use of robot motion planning algorithms to generate collision-free paths. These algorithms typically divide the desired path into a series of waypoints. Furthermore, given an initial time t_s and a final time t_f for the trajectory, offline trajectory planning

techniques can be utilized. These techniques establish the desired velocity profile along the path. This profile is represented by the EE linear velocity, $\dot{\boldsymbol{p}}_{EE,d}(t)$, and angular velocity, $\dot{\boldsymbol{\varphi}}_{EE,d}(t)$, for any time instant t between t_s and t_f. This approach empowers the robot to navigate towards its designated target position while ensuring collision avoidance with obstacles present in the environment. Once the trajectory is defined in the operational space as $\boldsymbol{\chi}_{EE,d}(t) = [\boldsymbol{p}_{EE,d}(t)\ \boldsymbol{\varphi}_{EE,d}(t)]^T$, the value of the velocities in joint space must be determined. A possible solution is the inverse kinematics algorithm with Jacobian inverse. Therefore, the joint velocity vector is:

$$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})(\dot{\boldsymbol{\chi}}_{\boldsymbol{d}} + \boldsymbol{K}_J \boldsymbol{e}) \tag{5.5}$$

where $\boldsymbol{q}$ is the 6×1 joint position vector, J^{-1} the 6×6 Jacobian inverse, $\boldsymbol{K}_{\boldsymbol{J}}$ is a positive definite 6×6 matrix, and $\boldsymbol{e}$ is the pose error in the operational space, $\boldsymbol{e} = \boldsymbol{\chi}_d - \boldsymbol{\chi}_{EE}$. When the obstacle enters the workspace of the robot, it can interfere with the path of the robot. Collision avoidance algorithms address this by incorporating a repulsive velocity term into the robot control law, that depends on the distance between the robot and the obstacle. Defined the distance between the robot and the obstacle as:

$$\boldsymbol{d}_{EE} = \boldsymbol{p}_{EE} - \boldsymbol{p}_o \tag{5.6}$$

the linear repulsive velocity is defined as:

$$\boldsymbol{v}_{rep} = v_{rep} \frac{\boldsymbol{d}_{EE}}{||\boldsymbol{d}_{EE}||} \tag{5.7}$$

with

$$v_{rep} = \frac{v_{rep,max}}{1 + e^{(||\boldsymbol{d}_{EE}||(2/\rho_F)-1)\alpha_F}} \tag{5.8}$$

Here, v_{rep} is the magnitude of the repulsive velocity, $v_{rep,max}$ the maximum value at the repulsive velocity, and α_F and ρ_F are two parameters that define the curve of the repulsive velocity. In particular, α_F determines the steepness of the curve and ρ_F gives an indication of the value of the maximum distance after which v_rep is active. The total repulsive velocity vector is then composed as:

$$\dot{\boldsymbol{\chi}}_{rep} = \begin{bmatrix} \boldsymbol{v}_{rep} \\ \boldsymbol{0} \end{bmatrix} \tag{5.9}$$

in which $\mathbf{0}$ is the null vector of dimensions 3×0, and stands for the angular velocity vector. The velocity control law is then:

$$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})(\dot{\boldsymbol{\chi}}_d + \boldsymbol{K}\boldsymbol{e} + \dot{\boldsymbol{\chi}}_{rep}) \tag{5.10}$$

This strategy considers only the distance of the EE from the obstacle, and only the EE is affected by the repulsive action, due to the fact that the complete Jacobian is considered. However, this approach can be extended to the whole robotic arm.Several control points can be identified on the robot. Simultaneously, the robot is divided into its links, and for each link, a group L_i of the relative control points is defined, where i stands for the i_{th} link of the robot. Therefore, the distance of the obstacle from each control point can be calculated. For each link, the minimum distance to the obstacle can be determined by finding the minimum distance d_{L_i} among all the control points in L_i. Therefore, for each link a repulsive velocity v_{rep,L_i} can be defined and applied to the point of minimum distance:

$$\boldsymbol{v}_{rep,L_i} = v_{rep,L_i} \frac{\boldsymbol{d}_{L_i}}{||\boldsymbol{d}_{L_i}||} \tag{5.11}$$

$$v_{rep,L_i} = \frac{v_{rep,max}}{1 + e^{(||\boldsymbol{d}_{L_i}||(2/\rho_F)-1)\alpha_F}} \tag{5.12}$$

Thus, the total repulsive vector is:

$$\dot{\boldsymbol{\chi}}_{rep,L_i} = \begin{bmatrix} \boldsymbol{v}_{rep,L_i} \\ \mathbf{0} \end{bmatrix} \tag{5.13}$$

The repulsive velocity can be transformed in the joint space velocity, considering the inverse of the partial Jacobian $J_{L_i}^{-1}$:

$$\dot{\boldsymbol{q}}_{L_i} = J_{L_i}^{-1} \dot{\boldsymbol{\chi}}_{rep,L_i} \tag{5.14}$$

The partial Jacobian J_{L_i} is calculated in the point of minimum distance of the group L_i. The velocity control law becomes:

$$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})(\dot{\boldsymbol{\chi}}_d + \boldsymbol{K}\boldsymbol{e}) + \sum_{L_i} \dot{\boldsymbol{q}}_{L_i} \tag{5.15}$$

in which the sum considers the contributions of each link repulsive action on the joints.

Within collision avoidance applications, simplifying the real shapes of robots and obstacles with basic geometries offers several advantages. Collaborative robotics, for instance, frequently employs cuboids, ellipsoids, or cylinders to represent the approximate dimensions of humans and robots. This approximation allows to simply calculate the distances between robot and operator, and to inflate real volumes to increase safety in the operations. In the case of the human operator, body parts as arms and legs are described by capsules, i.e., cylinders with spheres at their extremities. The radius of the spheres can be greater or equal than the radius of the cylinders, and their centers must be defined on the axis of the cylinder [131]. The distance between the capsule and an external point can be calculated defining the reference frame $O_c - x_c y_c z_c$, placed on one end of the cylinder. In this reference frame the x_c axis is aligned with the axis of the cylinder, the x_c axis points upward and y_c axis is given by the right-hand rule. Consider a point A outside the capsule, figure 5.12. The position of the center of the spheres in the $O_c - x_c y_c z_c$ reference frame are ${}^c\boldsymbol{s}_1$ and ${}^c\boldsymbol{s}_2$, and the radius are r_1 and r_2, respectively. The distances from the point A and the spheres, i.e., the external surface if the spheres, are:

$$d_{r_1} = {}^c\boldsymbol{p} - {}^c\boldsymbol{s}_1 - r_1 \frac{{}^c\boldsymbol{p} - {}^c\boldsymbol{s}_1}{\|{}^c\boldsymbol{p} - {}^c\boldsymbol{s}_1\|} \tag{5.16}$$

$$d_{r_2} = {}^c\boldsymbol{p} - {}^c\boldsymbol{s}_2 - r_2 \frac{{}^c\boldsymbol{p} - {}^c\boldsymbol{s}_2}{\|{}^c\boldsymbol{p} - {}^c\boldsymbol{s}_2\|} \tag{5.17}$$

where ${}^c\boldsymbol{p}$ is the position vector of point A. The distance from the cylinder instead is:

$$d_{r_3} = {}^c\boldsymbol{p}_{y_c z_c} - {}^c\boldsymbol{s} \tag{5.18}$$

in which ${}^c\boldsymbol{p}_{y_c z_c}$ is the projection of ${}^c\boldsymbol{p}$ in the $y_c z_c$ plane, and ${}^c\boldsymbol{s}$ the point on the surface of the cylinder that is closer to the point A.
The value of the minimum distance of the capsule from the point is:

$$d_r = \begin{cases} \min(d_{r_1}, d_{r_2}, d_{r_3}), & if \quad 0 \leq {}^c p_x \leq h_c \\ \min(d_{r_1}, d_{r_2}), & if \quad {}^c p_x < 0 \quad or \quad {}^c p_x > h_c \end{cases} \tag{5.19}$$

A similar approximation can be made with the robot: each link is surrounded by a series of spheres, centered in the control points, in order to cover the entire robot. The distance between each sphere of the robot and each capsule is identified. The

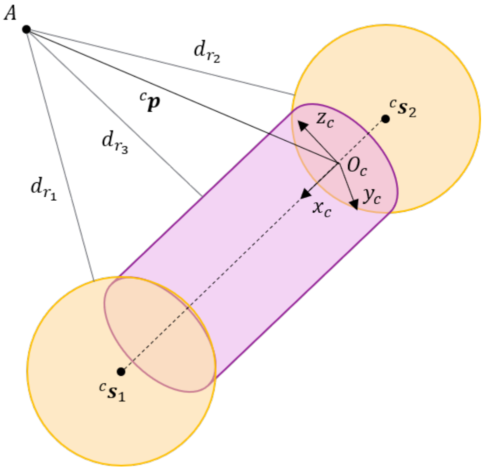


Fig. 5.12 Example of capsule, with external point, for distance calculation.

value is equal to the one defined in 5.19, to which is subtracted the radius of the sphere on the robot. The minimum distance for each group L_i is individuated as d_{L_i}.

5.3 Simulation tool

A simulation tool has been developed for testing and validating the robotic cell performance before deployment in a real-world environment. Here, different scenarios can be simulated to refine the control algorithms, optimize the workflow, and ensure safety.

The simulation environment is built using CoppeliaSim, where the detailed kinematic models of the robots, the static model of the battery being disassembled, and the kinematic model of a human operator represented by a full mannequin are developed. This environment allows for the visualization of the robot interactions with the environment and the human operator, ensuring that the control algorithms can be tested under realistic conditions.

Matlab is used to calculate the trajectories followed by both robots. The Matlab code determines the paths for the robots, and this information is integrated with CoppeliaSim for detailed simulation. In addition, the motion of the operator, which is previously captured using kinect cameras, is sent to the simulation environment, allowing the mannequin to move as a real operator. This integration allows for control and monitoring of robots actions. In figure 5.13, the simulation environment built in CoppeliaSim is shown, complete with the robots Franka Fr3 and Tiago Base, the mannequin and the table on which the battery is placed. With this simulation tool, the collision avoidance algorithm can be tested.

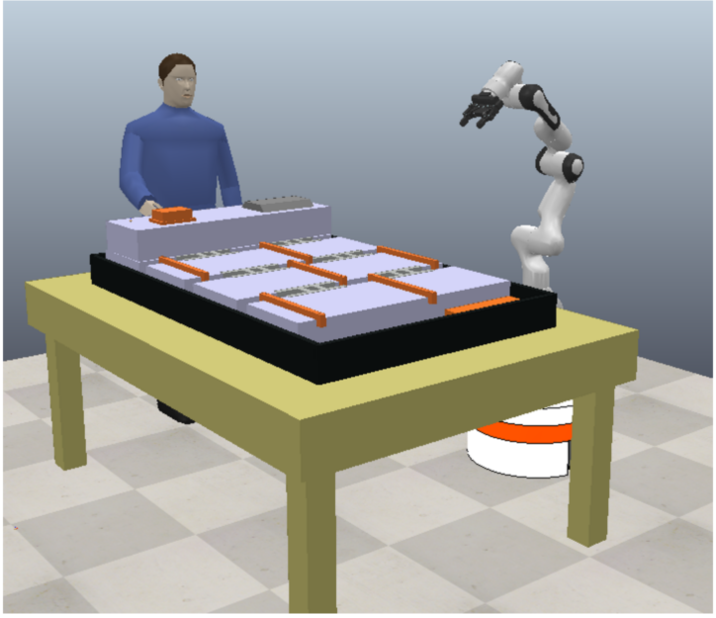


Fig. 5.13 CoppeliaSim environment.

Both the collision avoidance on the Franka robot and on the Tiago base are implemented.

5.3.1 Tiago robot collision avoidance

In developing the guidance algorithm for the mobile base, the system must be capable of navigating the base to the target working position defined in the disassembly cycle. It must also correctly orient the base while preventing any collisions with the human operator and the immediate environment, i.e., the work surface holding the battery. The collision avoidance algorithm is the same introduced in section 4.2.2, but the presence of the table has been included. The operator has been surrounded by a cylinder of 0.4 *m* diameter, whose center is aligned with the mannequin torso. The table, instead, is defined as a sigmoidal surface that includes it.
The virtual potential field associated to the table is defined as:

$$U_t = K_t \prod_{i=1}^{4} \frac{1}{1 + e^{-\gamma_t (A_i(x_{mr} - x_t) + B_i(y_{mr} - y_t) + C_i)}} \tag{5.20}$$

in which K_t is a coefficient that defines the intensity of the field and γ_t its decay, A_i, B_i and C_i allow to define the rectangular shape of the table, with $i = 1,2,3,4$ that refers to the sided of the rectangle, x_{mr} and y_{mr} are the coordinates of the mobile robot, and x_t and y_t are the coordinates of the center of the table. For details, refer to [132]. In the simulation tool, $K_t = 1$ and $\gamma_t = 17$. The dimensions of the rectangle are $2.54m \times 1.94m$. The resulting potential field is shown in figure 5.14. As the mannequin does not move in the space, but moves only its right arm, the local attractor is not considered. The total potential field is then:

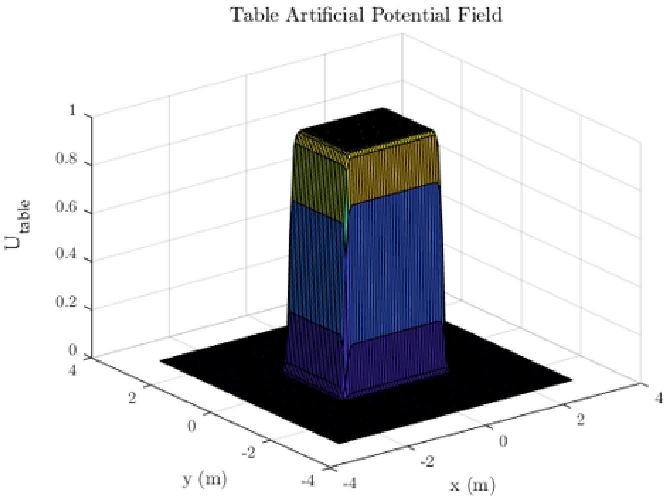


Fig. 5.14 Artificial potential field of the table.

$$U_{tot} = U_t + U_a + U_m \tag{5.21}$$

with U_a the global attractor potential field, and U_m the repulsive potential field of the mannequin. At each instant, the desired direction of the mobile robot is:

$$\boldsymbol{\lambda_d} = -\frac{\nabla U_{tot}}{\|\nabla U_{tot}\|} \tag{5.22}$$

The velocity of the robot can be seen as the sum of two contributions, one attractive $\boldsymbol{v}_a = -\nabla U_a$, and one repulsive $\boldsymbol{v}_r = -\nabla(U_t + U_m)$. It is essential that the base can navigate towards the target working position specified in the disassembly cycle. Additionally, it must orient itself correctly while avoiding collisions with the human operator and the immediate environment, specifically the table holding the battery. The algorithm must be improved to optimize the path when multiple options are available and to prevent the mobile base from stopping at an undesired position due to a local minimum in the overall potential field, which would result in a zero velocity.

The path selection is based on global navigation considerations: since the mobile robot must reach positions around a table, the primary movement involves circumnavigating the space occupied by the table. This movement can be performed either clockwise or counter-clockwise. By considering the current position of the base and the target position, it is straightforward to evaluate which path is shorter. The angle between the segment connecting the mobile base center to the center of the working table and the segment connecting the target position to this point indicates whether the clockwise or counter-clockwise path is shorter. Simplifying the trajectory as circular, the robot should follow the path that forms an arc subtending an obtuse angle. This directional reference is maintained throughout the navigation to determine the base speed at each instant, guiding the mobile base to the target position in the absence of other obstacles along the trajectory. Local minimum points are addressed by considering the special case where the velocity for the moving base, calculated based on the gradient of the potential fields, is zero. This situation arises when $\boldsymbol{v}_a = -\boldsymbol{v}_r$. In such cases, the ratio $r = \frac{\boldsymbol{v}_a}{\boldsymbol{v}_r}$ is defined; at local minima, this ratio is $r = 1$. Moreover, the versors λ_a and λ_r of the velocities $\boldsymbol{v}_a$ and $\boldsymbol{v}_r$ respectively are such that $\lambda_a \cdot \lambda_r = -1$. A new versor λ_t is introduced, perpendicular to both λ_a and λ_r. Since the movement of the mobile base occurs in a plane, there are only two possible directions to select from. The most appropriate direction is chosen based on the global navigation considerations mentioned above. Denoting by φ the angle described to circumvent the obstacle in a counterclockwise direction, the matrix M is defined as:

$$\boldsymbol{M} = \begin{cases} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \text{if } \varphi \leq \pi \\ \\ \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } \varphi > \pi \end{cases} \tag{5.23}$$

Then, it is determined:

$$\vec{\lambda}_t = \boldsymbol{M}\vec{\lambda}_r \tag{5.24}$$

For values of r close to 1, the versor λ identifying the direction of the moving base velocity is defined as:

$$\lambda = \frac{r\lambda_t + (1-r)\lambda_a}{|r\lambda_t + (1-r)\lambda_a|} \tag{5.25}$$

5.3.2 Franka robot collision avoidance

The collision avoidance between the Franka FR3 robot and the mannequin follows what explained in section 5.2. The robot is consider as a set of fifteen spheres, placed along its body, figure 5.15. Each sphere is associated to a link, for the definition of the repulsive velocity of the links. In table 5.8, the map between spheres and links is reported. In the simulation tool, the robot is reduced to a 6-dof robot imposing $q_7 = 0$ during the simulation. The value of α_F and ρ_F are $\alpha_F = 6$ and $\rho_F = 0.25$.

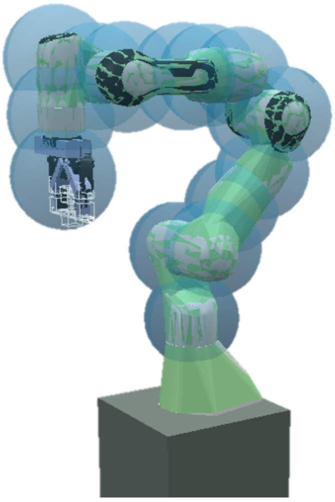


Fig. 5.15 Simulation Franka FR3, with the fifteen spheres for the collision avoidance.

Sphere	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Link	1	1	2	2	3	3	4	4	4	5	5	6	6	6	6

Table 5.8 Table linking spheres to links

The resulting correction coefficient for v_{rep,L_i}, normalized with respect to $v_{rep,max}$ is the one shown in figure 5.16.

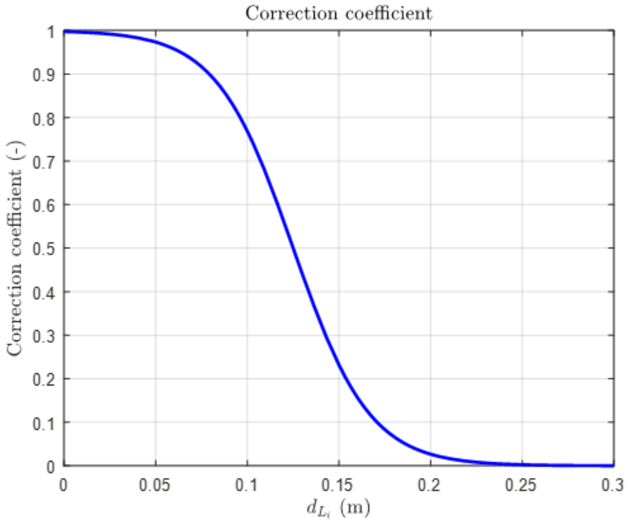


Fig. 5.16 Correction coefficient for repulsive velocity, for the simulated Franka robot.

5.3.3 Results

The algorithms are evaluated in a simulation environment to verify their effectiveness and assess their capability to generate trajectories that avoid collisions with the human operator while reaching the designated position for the planned dismantling operation. Two typical scenarios are considered:

a) The mobile base follows a path that intersects the area occupied by the operator and concludes its movement in a zone where the robotic arm operates, potentially interacting with the human operator.

b) The mobile base remains in a fixed position while the robotic arm carries out a disassembly task, sharing the workspace with the human operator.

The results of the tests can be visualized in the simulation environment. In figure 5.17, some instants of the test (a) in CoppeliaSim enviroment are shown. The robotic manipulator moves from its starting position to the target one, moving around the table and the operator, and the robotic arm reaches its final position on the working table. The two robot moves according to the algorithms shown in the previous sections.

In Figure 5.18, the motion of the Tiago within the environment is depicted. The starting position is indicated by a dark gray circle surrounded by a green

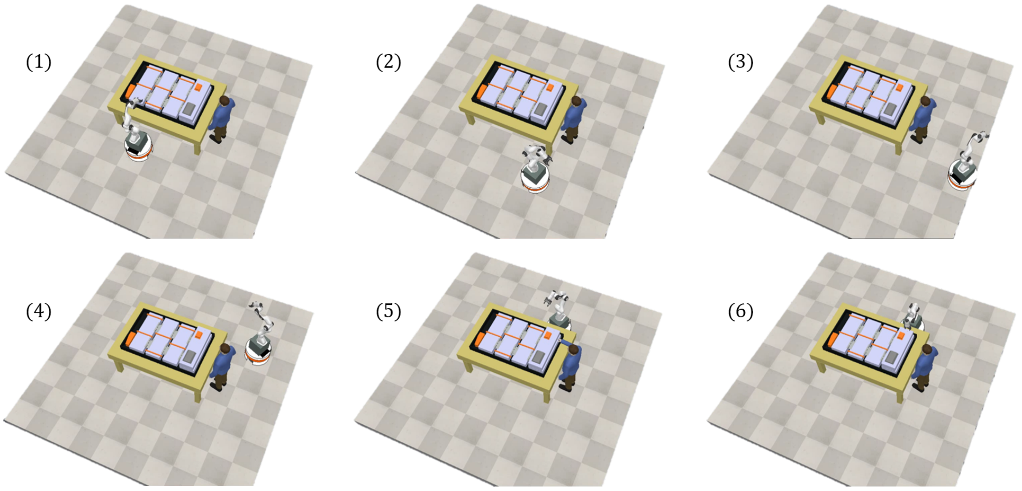


Fig. 5.17 Six instants of test (a), in CoppeliaSim environment.

circumference, while the target position is marked by a dark gray circle encircled by a magenta circumference. The operator's location is shown as a red circle, and the table is represented by a purple rectangle. The robot trajectory is illustrated with a blue line, and the areas occupied by the robot at various instances are shown as light gray circles. In a first step, the algorithm recognizes the opportunity to follow a counterclockwise path to circumvent the table. Then, it detects the presence of the operator in its path and deviates its trajectory to avoid him, passing behind him.

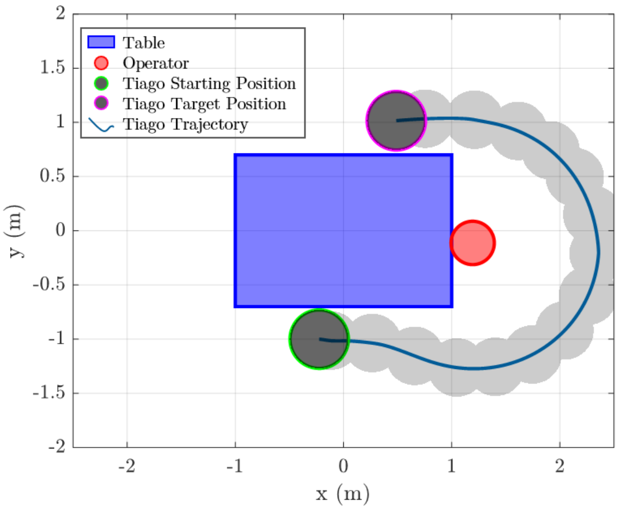


Fig. 5.18 Trajectory of the Tiago robot in the environment, during test (a).

In figure 5.19, eight frames depicting the motion of the Franka robot around the table are shown.

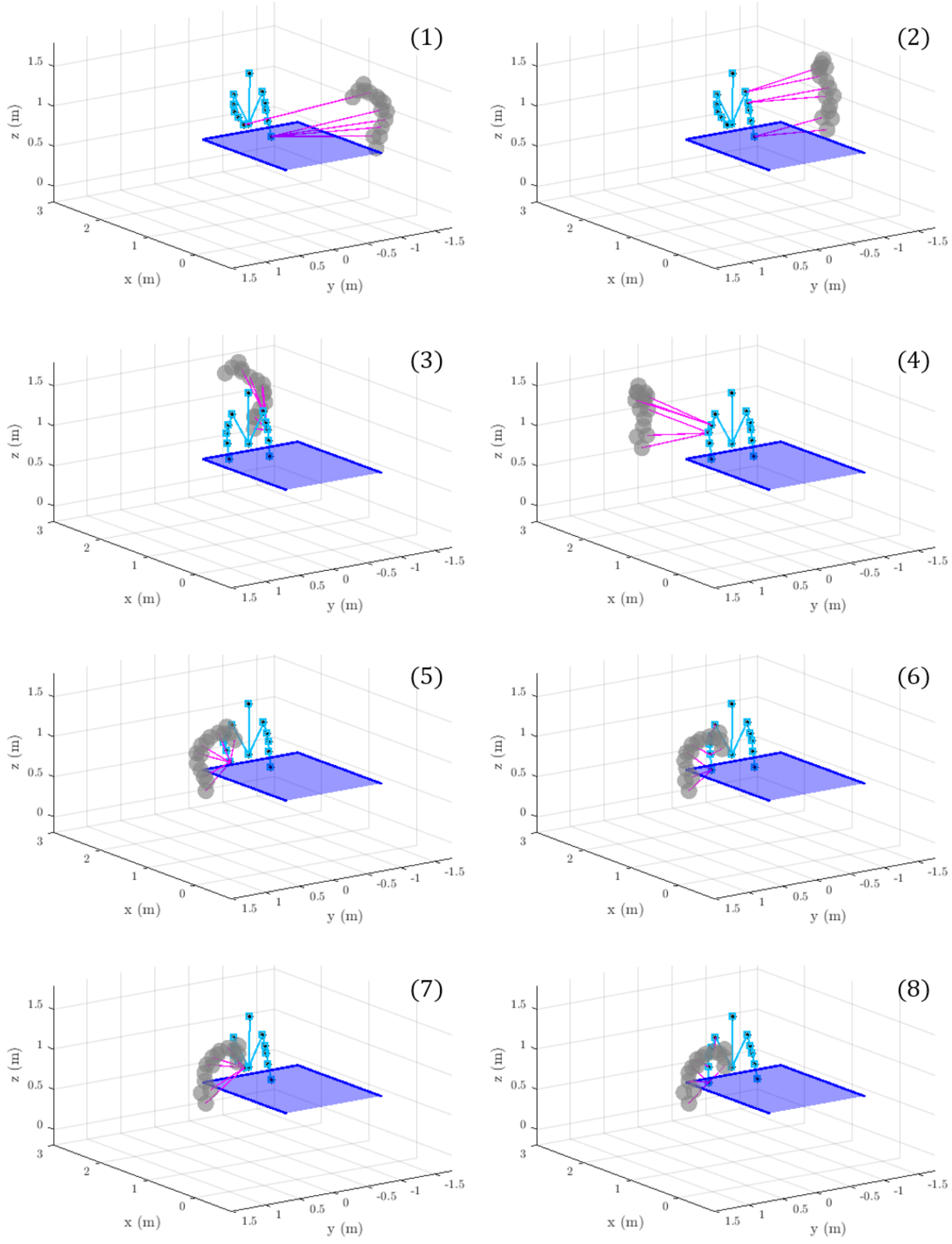


Fig. 5.19 Eight instants of test (a), with the Franka robot, the operator and the table. Minimum distances are highlighted.

The robot is represented by gray spheres which correspond to the spheres placed on the robot, as shown in figure 5.15, for the collision avoidance of the robot. The operator is shown as a light blue skeleton, with the points that correspond to the places where the interest points for collision avoidance are placed. The working table is shown in blue. The minimum distances of each link group of the robot and the operator are highlighted in magenta. As can be seen, the algorithm checks for the collision distance from the Franka and the operator during all the simulation. In Figure 5.20, five frames from test (b) in the CoppeliaSim environment are depicted. In this test, the Tiago robot remains stationary while the EE of the Franka robot moves to reach a point on the worktable. However, as the mannequin moves, its right arm intercepts the robot path. Consequently, the robot deviates from its trajectory to avoid the collision and then returns to its original path. This can be seen in figure 5.21, in which two overlapped instants of the simulation are shown. The first one in shaded blue, before the operator moves the arm, and a second one, in shaded red, in which the operator have the arm extended, and the robot moves to deviate it.

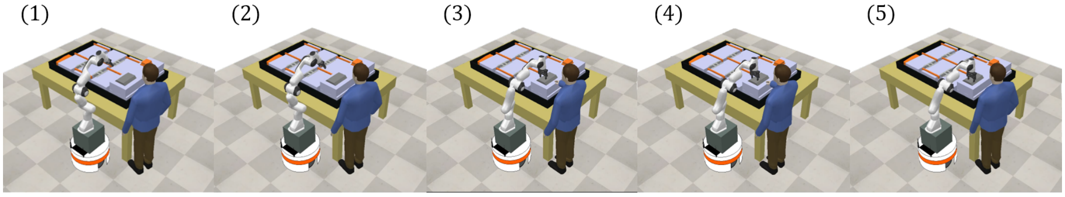


Fig. 5.20 Five instants of test (b), in CoppeliaSim environment.

The developed simulation tool has proven to be an excellent resource for testing the algorithms before implementing them in a real-world environment. It allows for thorough evaluation and refinement of the algorithms, ensuring their effectiveness and reliability prior to conducting experiments with the actual robots. This approach not only minimizes the risk of errors during live testing but also provides valuable insights into potential improvements. By simulating various scenarios and conditions, the tool helps identify and address issues that may not be apparent in a real-world setup, ultimately enhancing the overall robustness and performance of the robotic system.

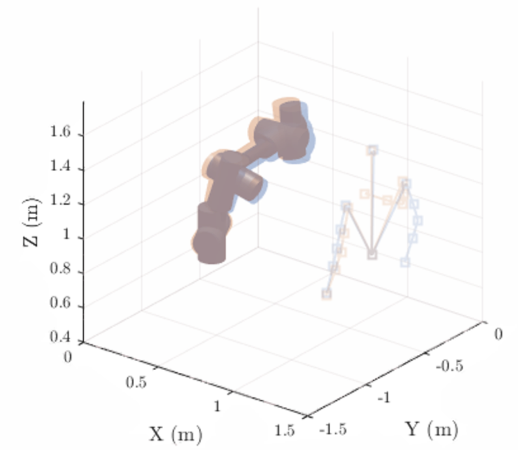


Fig. 5.21 Two instants of test (b), with the Franka robot, the operator and the table. The first instant is in blue, with the operator arm along the body, and the second one in red, with the extended person arm, and the robot that deviates it.

5.4 Experimental tests

The experimental tests conducted to validate the functionality and safety of the collaborative robotic cell are presented. The tests were designed to evaluate the system performance in three key areas: collision avoidance between the Tiago robot and the human operator, precise positioning of the Franka robot relative to the workbench, and collision avoidance between the robotic arm and the human operator.

5.4.1 Experimental tests: Tiago robot and operator

The first set of tests focused on validating the collision avoidance capabilities of the Tiago mobile base in the presence of a human operator. For these tests, the Franka Emika robotic arm was not active.

The most critical interaction identified when using the mobile robot within the workplace, particularly in the broader industrial environment, occurs when the robot encounters a human operator. Research indicates that if the human operator can predict the direction in which the robot will move, their comfort and confidence in interacting with the robot are significantly enhanced [115, 133]. Therefore, developing an algorithm that makes the robot movement direction predictable to humans is of considerable importance and has significant potential applications in the industrial context. So, the primary objective of this test was to validate the collision avoidance capabilities of the Tiago mobile base in the presence of a human operator.

The algorithm used in these tests is the same as the one described in section 4.4.3, except that the robot pose is no longer tracked using a camera and Aruco markers. Instead, the pose data is obtained directly from the robot SLAM (Simultaneous Localization and Mapping) system. SLAM is a technique used in robotics and autonomous vehicles that allows a robot to create a map of its environment while simultaneously keeping track of its location within that map. The robot creates the environmental map using the *gmapping* algorithm [134] and then uses Adaptive Monte Carlo Localization (AMCL) filtered particles to estimate its pose within the space [135, 136]. The SLAM algorithm implemented on the robot is provided by the manufacturer. The output of the SLAM process is the robot pose relative to the reference frame of the map created by the robot, that can be read as a message from the ROS topic $/robot\_pose$.
Before starting with the tests, a mapping of the laboratory was conducted. The result of the mapping is shown in 5.22, in which also the map reference frame is present. The robot also detected the furniture present in the lab, and because of the presence of stuff, it cannot complete the top left and the bottom right angles of the map. However, the map was sufficiently accurate for the purposes of the tests.

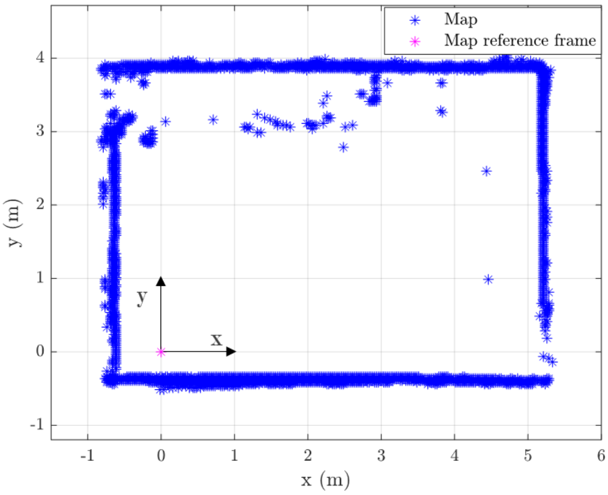


Fig. 5.22 Map of the laboratory with its reference frame.

Experimental Setup

In the laboratory, Kinect sensors were placed to track the movement of the person. The Kinects were positioned to capture the person walking from behind. To initialize the transformation matrix between the Kinect and the Tiago base ${}^{map}\boldsymbol{A}_K$, the solid markers, i.e., the three cones, were placed on the structure built on the robot, figure 5.23. A reference frame was defined on one of the three markers, making it the reference system for the Kinect. Since the reference frame of the Tiago is located at the center of the robot, the Kinect reference system was translated to the center of the Tiago, which is positioned along the diagonal of the rectangle formed by the markers, specifically the distance between the two white cones in the figure. Thus, by reading the position of the robot in the map, the transformation matrix between the Kinect and the map was calculated.

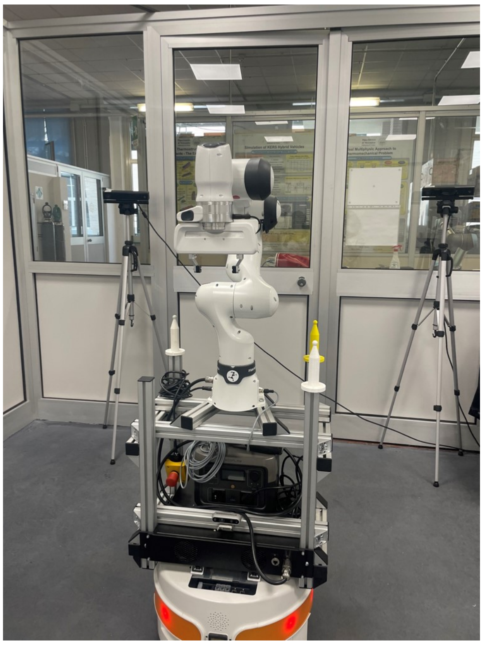


Fig. 5.23 Setup for the identification of the transformation matrix Kinect-Map.

The skeleton tracked by the Kinects is processed on two computers, a master and a slave, which are connected to the computer that controls the robot via a switch. The master PC sends skeleton data as a ROS topic that can be read by the algorithm on the Tiago computer. Only the spine joint position detected by the Kinect is sent, as the person is treated as a moving circle, with the radius being the sum of the radii of both the person and the robot.

Modified algorithm

The code was modified from the previous version because the robot would not be able to avoid the person in time. The first modification concerns the meaning of the obstacle circle R_o. Previously, it represented the combined dimensions of the person and the robot. Now, it serves as a tool to construct the potential field in which the robot moves. After an experimental tuning, the value chosen for the tests is $1.7m$. The value of the local attractor is related to the radius of the obstacle. The second modification involves adjusting the value of the local attractor radius based on its position relative to the global attractor. The local attractor now moves in the space together with the moving person. When the distance between the local attractor and the global attractor is less than the radius of the local attractor, the radius of the local attractor is set to be equal to the distance between the two attractors. This can be expressed mathematically as:

$$R_a = \min(R_a, d_{local-global}) \tag{5.26}$$

where $d_{local-global}$ is the distance between the local attractor and the global attractor. This adjustment ensures that the potential field around the local attractor is dynamically resized to prevent conflicts when the local attractor is close to the global attractor.

The third and final modification concerns the calculation of the robot linear velocity. Since the robot was found to be unable to turn in time when a person approached, it was decided to limit the linear velocity to allow the robot to change direction at a greater distance from the approaching person. To implement this, the position the robot would occupy after a certain number N of samples is predicted using the current value of the robot velocity v_T, if the robot were moving only with linear velocity:

$$\boldsymbol{x}_{T,future} = \boldsymbol{x}_T + \boldsymbol{\vartheta}_T\, v_T \cdot \frac{N}{f} \tag{5.27}$$

in which $\boldsymbol{x}_T$ is the actual pose of the Tiago robot, $\boldsymbol{\vartheta}_T$ is the current direction of the robot and f is the control frequency. The velocity at the N_{th} sample, at time $t_N = t_i + N \cdot f$, is then calculated as the negative gradient of the field at the previously identified point $\boldsymbol{x}_{T,future}$, and the direction of this velocity $\boldsymbol{\vartheta}_{T,future}$ is determined. The direction error ε_{future} between the velocity calculated according to the previous algorithm, described in chapter 4, and the velocity at the N_{th} instant is calculated.

The velocity to be sent to the robot is then:

$$v_{T,set} = \frac{v_{T,setMax}}{1 + e^{\|\varepsilon_{future} - \alpha_T\| (\frac{2}{\rho_T}) \alpha_T}} \tag{5.28}$$

The values chosen for α_T and ρ_T are $\alpha_T = 3$ and $\rho_T = \frac{\pi}{4}$. The resulting correction coefficient, normalized for $v_{T,setMax}$ is shown in figure 5.24.

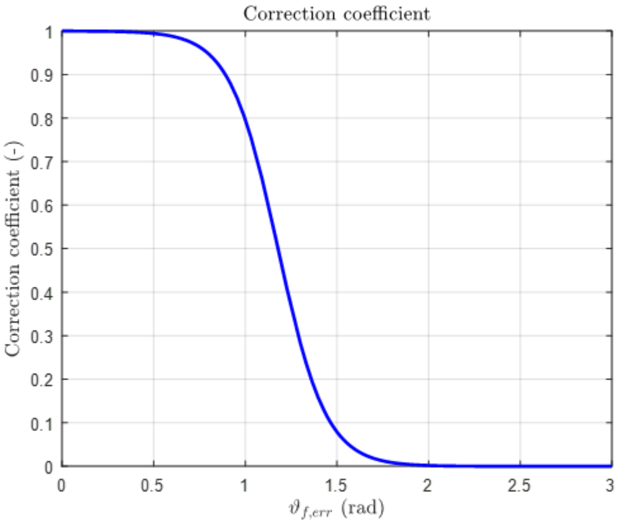


Fig. 5.24 Correction coefficient for the Tiago velocity.

The control frequency is set equal to 60 Hz, while the number of samples $N = 30$. The optimal values for the maximum linear velocity $v_{T,max}$ is set equal to $0.3m/s$ and the maximum angular velocity $\omega_{T,max}$ is set equal to 1.2 rad/s. Moreover, strength of the local attractor α_a is set equal to $\alpha_a = 0.99\tilde{\alpha}_a$, to enhance the attractive effect of the local potential field.

Results

The tests were conducted with the person walking directly towards the Tiago robot, moving in the opposite direction. During the tests, the person continued to walk straight without changing direction, to simulate the most extreme scenario and validate the effectiveness of the algorithm.

Three types of tests were performed. Three distinct test scenarios were executed to assess the system performance. In the first test, the person advanced towards the robot, and the local attractor was positioned to their left, offset by −45° from the vertical axis, positive upwards. This strategic configuration allowed the robot to detect the local attractor and subsequently the obstacle earlier, enabling it to initiate the avoidance maneuver sooner. In figure 5.25, four instants of the test are illustrated.

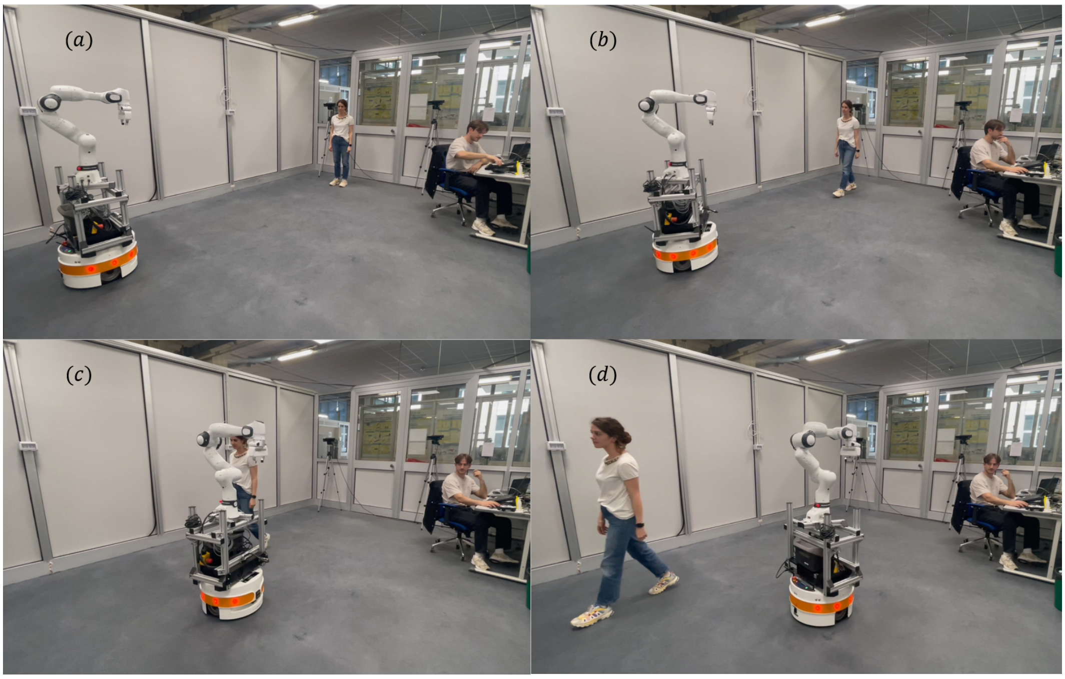


Fig. 5.25 Four instants of the test with Tiago robot, with the local attractor positioned on the left of the person.

Figure 5.25(a) shows the beginning of the test, figure 5.25(b) shows the moment in which the robot started rotating to avoid collision, figure 5.25(c) the moment when the person and the robot were approximately alligned on the $x-axis$, but were sufficiently distant to avoid collision, and figure 5.25(d) the moment after the person and the robot have passed by each other.

Figure 5.26 shows the plot of the distance between the robot and the person over time for Test 1. The plot shows in pink the safety distance between robot and person, that is the sum of the radii of the person and the robot. The radius of the robot was set to 0.32 m. While the average distance between the wrists of the recorded skeleton is approximately 0.21 m, the person's radius was set to 0.36 m to maintain a conservative estimate. This conservative approach ensures the test applicability to different individuals. The plot confirms that the robot and the person never hit during their motion.

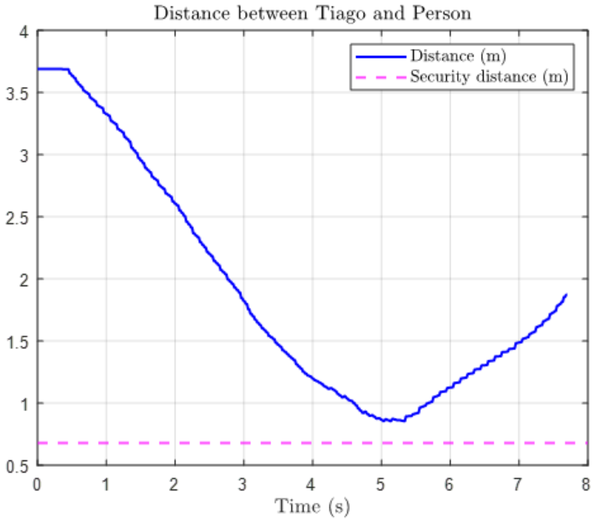


Fig. 5.26 Plot of distance versus time, test 1 of Tiago robot.

As the obstacle, i.e., the person, moves during the tests, the artificial potential field the robot moves in changes continuously. However, some instants of the artificial potential field isolines with the robot and the person are depicted in figure 5.27. In these figures, the robot starting position is marked with the yellow dot, the target position with a magenta dot, the robot position with a black dot. The repulsive potential field is depicted in red, and the attractive potential field is shown in green, while the circle relative to the safety distance is depicted in blue. The isolines are in blue, too. Figure 5.27(a) shows the start of tests. The local attractor is in the top left corner of the plot, and it exerts its influence from the beginning of the test, as can be noticed from the isolines. When the robot approaches the robot, figure 5.27(b), the isolines are vertical, to force the robot to turn. Figure 5.27(c) illustrates the moment of minimum distance between the robot and the person, with the robot escaping from the obstacle while following the isolines. Finally, figure 5.27(d), shows an instant when the person is no more detected by Kinect, after passing by the robot, and so the latter feels only the influence of the attractive potential field. The skeleton data recorded during Test 1 was subsequently utilized to conduct further tests, ensuring that the initial conditions remained consistent. The path of the person is shown in figure 5.28. The second test was conducted under the same conditions as Test 1, but with the local attractor positioned on the right side of the person, and the person motion is shifted in the positive direction of Y. The results of the test are shown in figures 5.29 and 5.30. As could be expected, the results are similar to test 1 and the

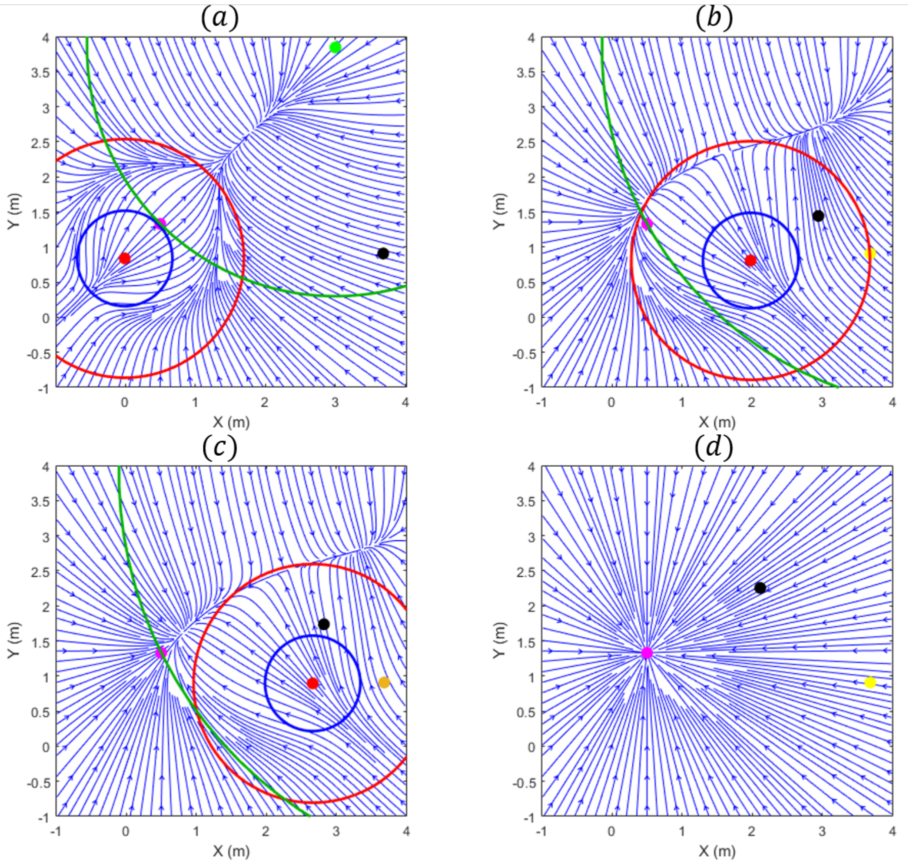


Fig. 5.27 Isolines of test 1 of Tiago robot, at different instants.

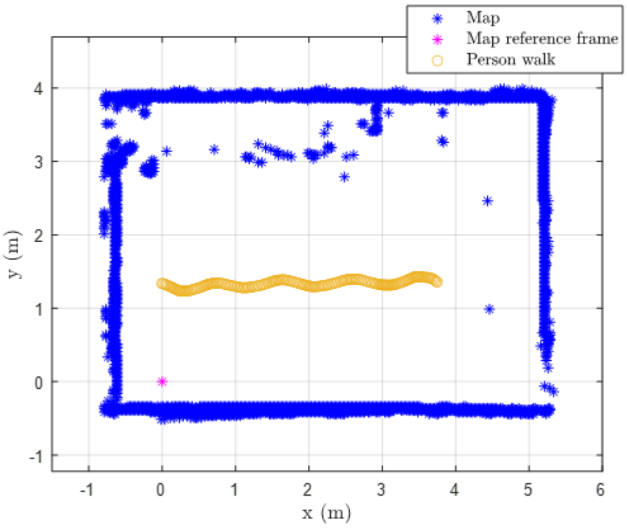


Fig. 5.28 Path of the person in the laboratory map.

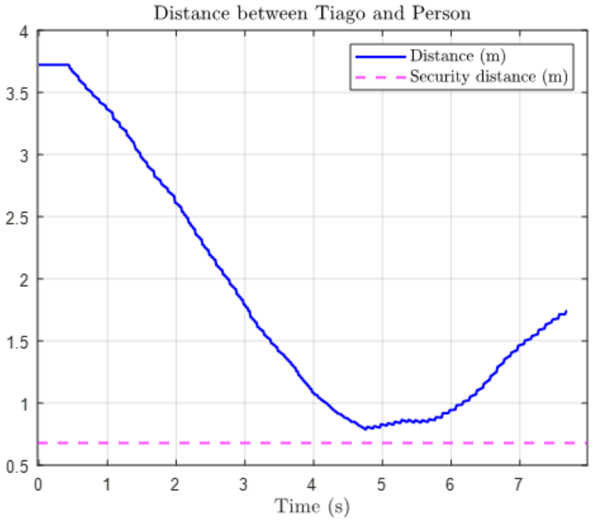


Fig. 5.29 Plot of distance versus time, test 2 of Tiago robot.

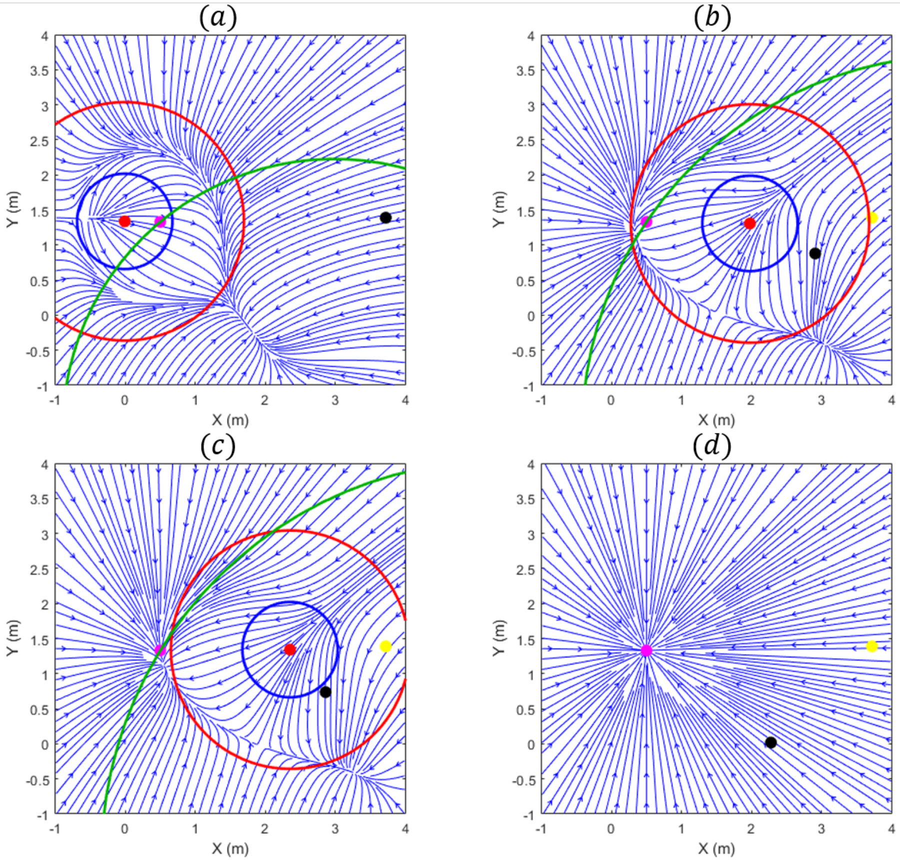


Fig. 5.30 Isolines of test 2 of Tiago robot, at different instants.

difference can be ascribed to the inaccuracy of the robot localization. However, the test confirms the good tuning of the parameters.
The results of test 3 are noteworthy. In this test, the same conditions were applied, but without the presence of a local attractor. As shown in figure 5.31, the robot fails to avoid the person and continues along its trajectory. The reason for this can be understood by examining figure 5.32, which displays the isolines at four different instances during the test. In figure 5.32(a), the initial moment of the test is shown.

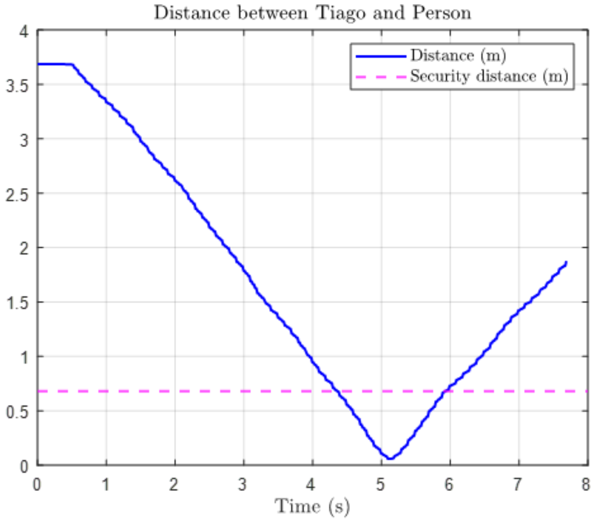


Fig. 5.31 Plot of distance versus time, test 3 of Tiago robot.

The absence of the local attractor, which in the previous tests was positioned ahead of the person, prevents the robot from receiving the information about the presence of the obstacle in time to initiate turning. Figure 5.32(b-c) show that, even as the robot approaches the obstacle, the repulsive potential field alone is insufficient to deviate the robot path. Consequently, in figure 5.32(d), the robot is entering the safety circle of the obstacle, failing the test.

5.4.2 Experimental tests: Spatial localization

The localization tests were designed to demonstrate that the setup described in the sections 5.1.3 - 5.1.4 allows the Franka robot to localize itself accurately near the table to perform precise tasks. Specifically, these tests show that the robot can position itself accurately enough to loosen a screw. In this particular test, the robot is

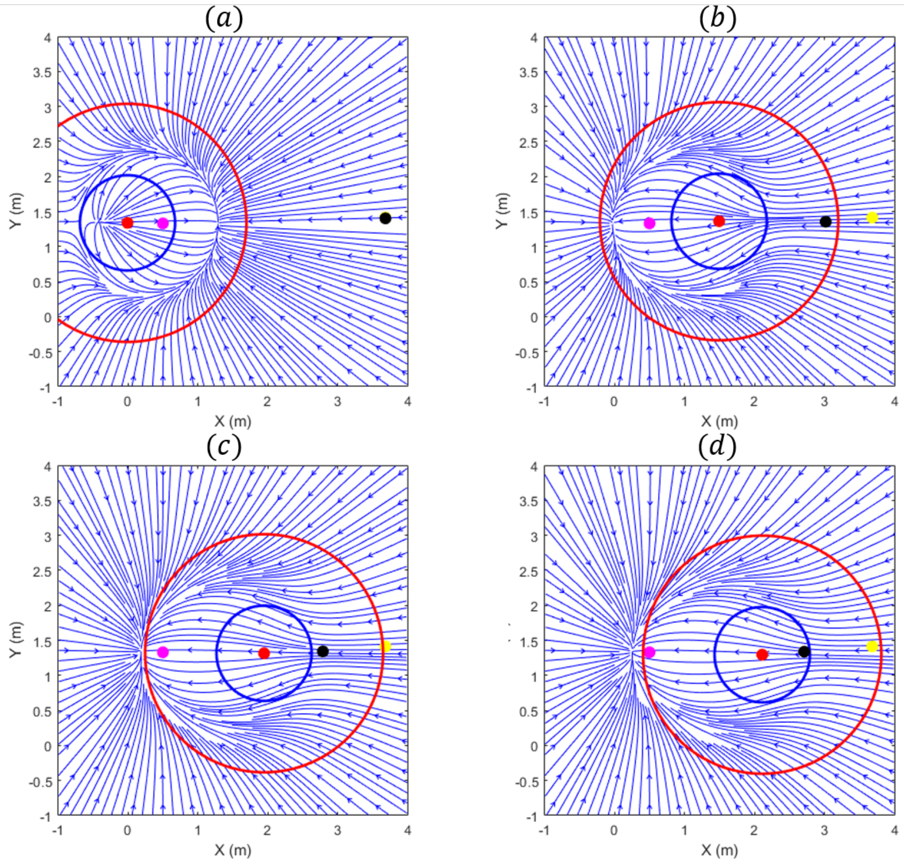


Fig. 5.32 Isolines of test 3 of Tiago robot, at different instants.

not equipped with a screwdriver, so instead of unscrewing, it simply positions itself over the screw, closes the gripper, and lifts the screw.

Experimental setup

For the tests, a new map of the laboratory was created, depicting a cleared space with the table accurately positioned, figure 5.33. This setup allows to detect the position of the table and to include it in the Tiago control algorithm. The Tiago and Franka computers are interconnected via a switch. When the Tiago robot reaches the desired pose near the table, it sends a trigger signal to initiate the Aruco marker pose reading by the RealSense camera. This data is then processed by the Franka robot controller, which uses the transformation matrices described in 5.1.4 to determine the EE precise pose to align with the screw. The setup of the test is represented in

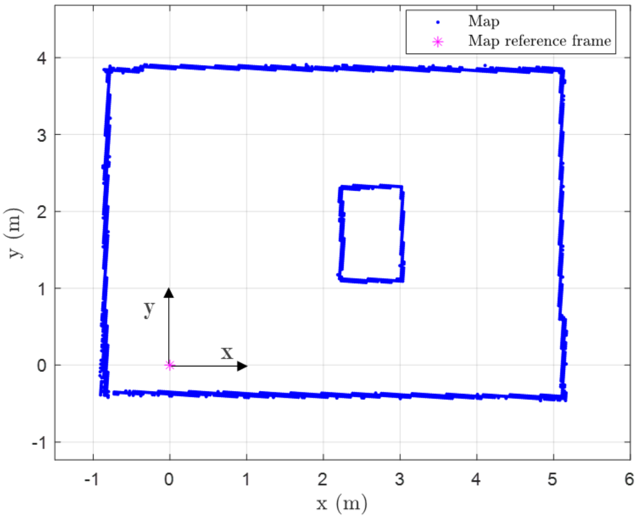


Fig. 5.33 New map of the laboratory, with the working table.

figure 5.11. During the tests, the Franka robot is connected via Ethernet cable to the PC.

Algorithms

To navigate the space, the Tiago robot employs the algorithm described in section 5.3.1, integrated with the modification for the collision avoidance with the walking person, presented in section 5.4.1. Upon reaching the target pose, it triggers the Franka robot to start the localization process. The Franka robot utilizes the localization algorithm discussed in 5.1.4, involving the reading of Aruco markers poses with the RealSense camera and the calculation of necessary transformations to accurately position the end effector over the screw.

Results

The results of the localization tests are presented in several figures. Figure 5.34 illustrates the path taken by the Tiago robot to reach the table. In the figure, the robot path is shown using the consecutive reference frames of the Tiago robot, with the axes x_T, y_T, and z_T represented in red, green, and blue, respectively. As can be seen, the robot approached the table, and then stopped to allow the Frank robot to perform its task.

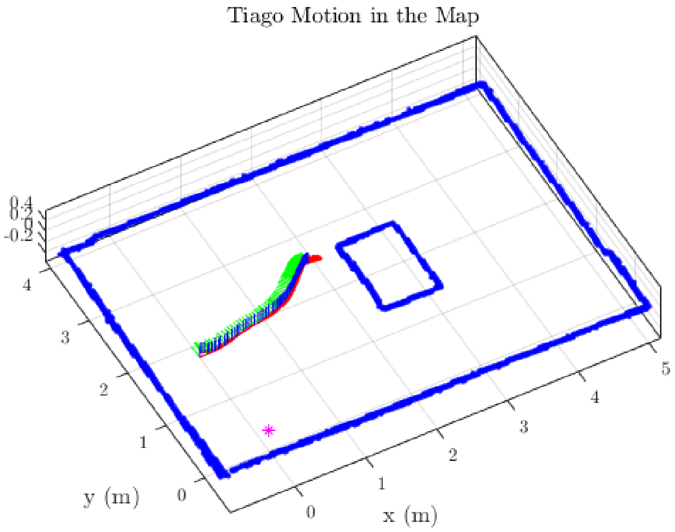


Fig. 5.34 Trajectory of the Tiago robot navigating to the table.

Figures 5.35, instead, depicts the Franka robot motion till the lifting of the screw. In particular, in figure 5.35(a) the robot is approaching the screw position, in figure 5.35(b) the robot closes its gripper and clamps the screw, and in figure 5.35(c) the robot takes the screw away. The sequence of movements showcases the robot ability to make very accurate adjustments based on the localization data from the Realsense camera and Aruco markers. This capability is crucial for tasks that require high precision, as taking a screw or unscrewing one.

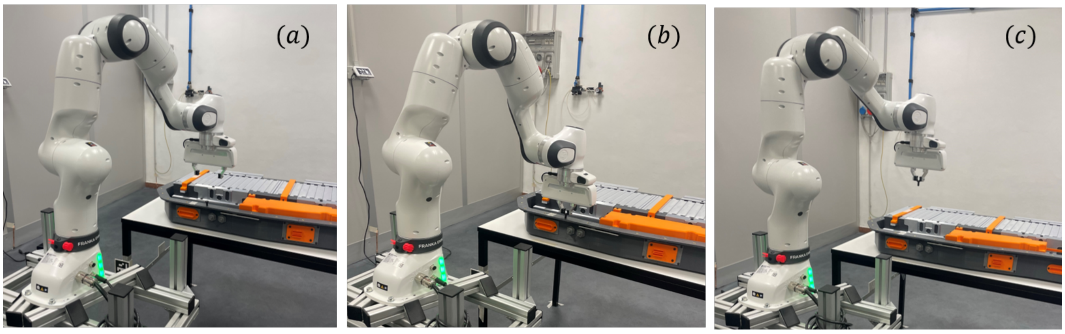


Fig. 5.35 Sequence of the Franka robot approaching the screw, and lifting it.

Finally, Figures 5.36 show the gripper closing around the screw and lifting it.

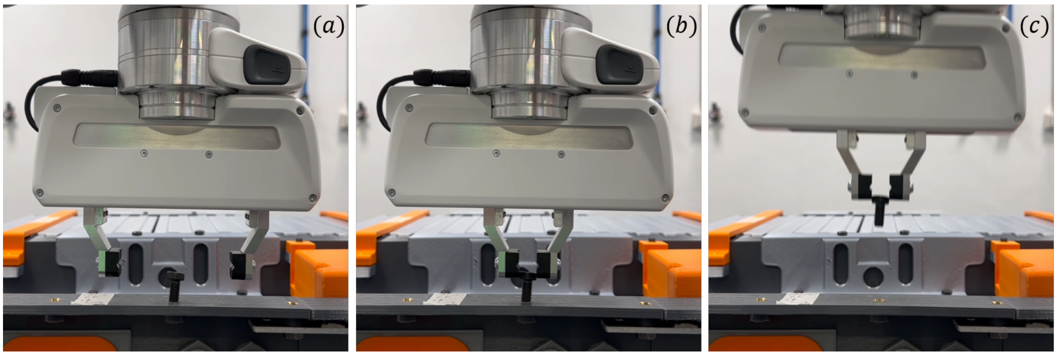


Fig. 5.36 Detail of the lifting of the screw.(a) the gripper reaches the screw, (b) closes its finger, and (c) the robot lifts the screw up.

Overall, the results from these tests confirm that the robotic system can accurately navigate and position itself within the environment, demonstrating its potential for performing complex and precise tasks. The combination of the Tiago robot navigation capabilities and the Franka robot precise localization results in a robust system capable of executing detailed and high-precision operations.

5.4.3 Experimental tests: Franka robot and operator

The objective of this test is to validate the effectiveness of the collision avoidance algorithm implemented on the Franka robotic arm, in the collaborative cell. Consistent with the methodologies discussed in chapter 2, the goal is to identify a task on the electric vehicle battery mockup that can effectively validate the collision avoidance algorithm. By analyzing video footage [61], it was determined that an efficient division of tasks between human and robot involves assigning the robot the task of unscrewing bolts, while the human operator handles more complex activities, such as disassembling the electrical components inside the orange box located at the front of the battery, and removing any components freed by the robot that may be stuck. Given the complexity of disassembling an electric vehicle battery, which involves various possible approaches, this division of labor was deemed a plausible situation. Thus, the objective is to demonstrate not only the robot ability to avoid collisions with the human operator but also to return to its designated work position to continue its assigned tasks.

Experimental setup

In the experimental setup, the human operator works on the orange box while the robot is tasked with unscrewing bolts from the battery module. The test scenario is designed to simulate a real-world environment where the human, having completed their task on the orange box, extends their arm to retrieve the busbar near the robot work area. This setup aims to test the robot ability to detect and avoid the human operator's movement and subsequently return to its task without interruption.
The experimental setup adheres to the configuration detailed in Sections 5.1.3 - 5.1.4, with the integration of Kinect sensors. To enhance spatial synchronization between the human operator and the Franka robotic arm, the transformation matrix between the Kinect and the Franka robot must be calculated. The reference frame for the Kinect sensors is defined as described in section 5.1.1. Figure 5.37 illustrates the setup, in which the solid markers, i.e., the cones, utilized to establish the Kinect reference frame are present. The Kinect reference frame is placed on the white cone in the corner. The transformation matrix between the Kinect reference frame and the Aruco markers ${}^{AM}\boldsymbol{A}_K$ on the table was accurately measured. This process enables

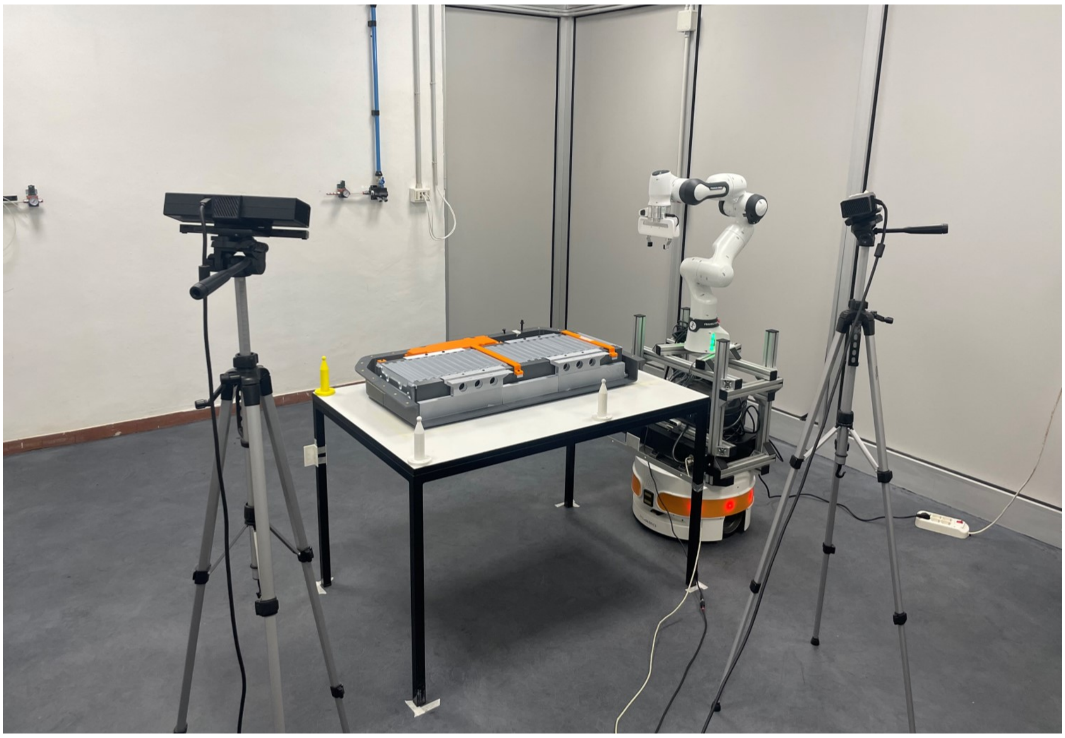


Fig. 5.37 Final setup with the solid markers used to define the Kinect reference frame.

the reconstruction of the transformation matrix between the Franka robot and the Kinect sensor ${}^{FR}\boldsymbol{A}^{K}$, following the methodology outlined in section 5.1.4:

$$ {}^{FR}\boldsymbol{A}^{K} = {}^{FR}\boldsymbol{A}^{RS}\,{}^{RS}\boldsymbol{A}^{AM}\,{}^{AM}\boldsymbol{A}_{K} \tag{5.29}$$

By reading the positions of the Aruco markers, the system can effectively synchronize the spatial coordinates of the robot and the person. The operator stands in from of the kinect cameras, as shown in figure 5.38. in the figure, the busbar is highlighted in the yellow oval, while the screw in the red one. The Kinect sensors are connected to two computers, a master and a slave. The Franka robot, its control computers, along with the master and slave Kinect computers, are connected via a network switch. The master computer processes the data and sends the human joint positions via a ROS message over a topic accessible to the Franka robot control computer.

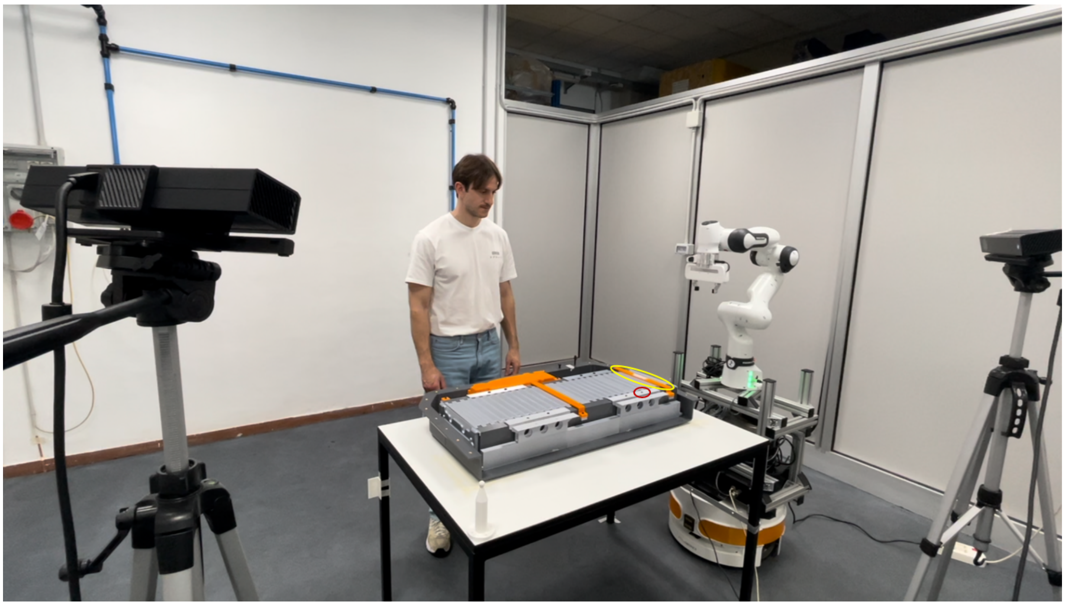


Fig. 5.38 Final setup with the operator.

Algorithm

The collision avoidance algorithm implemented on the Franka robot is the one presented in section 5.2. The algorithm relies on real-time data from the Kinect sensors and Aruco markers to dynamically adjust the robot movements. The Realsense information relative to the Aruco marker is handled as described in the previous

section. Overall, the combined algorithms for human detection, spatial synchronization, collision avoidance, and task execution enable the Franka Emika robotic arm to operate safely and efficiently.
In this phase of testing, the motion of the Tiago robot is not considered. The limited space available in the laboratory does not allow for extensive movements of the mobile robot within the collaborative cell setup. Therefore, the focus is solely on the Franka robot collision avoidance.

The task assigned to the Franka robot is to simulate the unscrewing of bolts from the EVB. Due to the absence of an actual screwdriver, the robot performs a simulated unscrewing process by positioning itself accurately over the bolt locations. Initially, the robot is in its home position. Then, the robot moves its EE from the home position to the position above the bolt. The localization of the bolt position is achieved using the RealSense camera and the Aruco markers. Once positioned above the bolt, the robot remains stationary for 12 seconds, to ensure the completion of the unscrewing operation. After, the robot lifts its EE vertically and then to its home position, completing the task cycle. During this process, the human operator may extend their arm to retrieve a busbar located in the robot workspace. The collision avoidance algorithm continuously calculates the proximity of the human operator's arm relative to the robot. According to the distance, the robot can temporarily halt its current action and move away from the operator's arm.
The safety distance after which the robot is subjected to a repulsive velocity is determined by equation 5.12, normalized by the value of $v_{rep,max}$. The parameter α_F is set to 6, while experimental tests were conducted with ρ_F values of both 0.33 and 0.4.
The robot is managed as a 6-dof robot by locking the third axis. The organization of the robot links into groups and the dimensions of the spheres placed along the robot are crucial for the collision avoidance algorithm. The robot is divided into 6 groups, each including control points. The dimensions of the spheres around the robot, in the control points, are selected to cover the entire volume of the robot links, ensuring comprehensive collision detection. All the spheres of the robot have radius of 0.1 m, but the last two on the Franka hand, which are equal to 1.2 m. The groups are organized as for the simulation tests, as described in table 5.8. Figure 5.39 shows the Franka robot with the spheres all around it. Each group is depicted with different color.

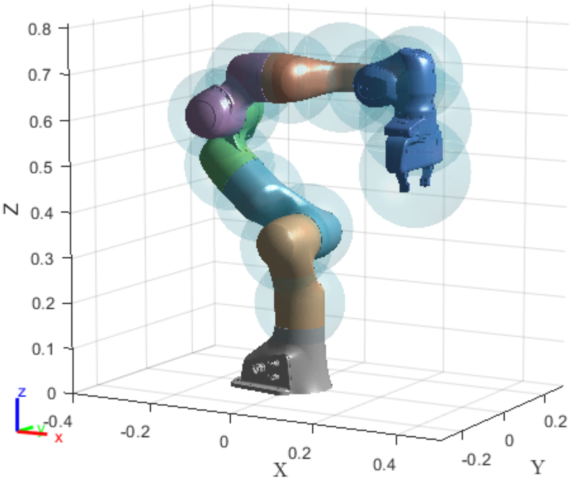


Fig. 5.39 Spheres for collision avoidance on the Franka FR3, for experimental tests.

Results

The tests were designed to evaluate the robot ability to perform its task while avoiding collisions with the human operator, ensuring both safety and task continuity. The robot behavior during these tests was influenced by the implementation of the collision avoidance algorithm, which is measures the distances between the robot and the operator during the whole task, and returns repulsive velocity when the a safety distance is breached.

Four types of tests were conducted to evaluate the collision avoidance algorithm under different conditions: two with $\rho_F = 0.33$ and two with $\rho_F = 0.40$. In each case, the algorithm was tested with the person extending their arm before the robot reached its task position and with the person extending their arm after the robot had positioned itself. In all four scenarios, the results were positive, demonstrating that the robot responded effectively to the human operator's movements. In the following,

- test 1 is the test with $\rho_F = 0.33$ and the person extending the arm before the robot reaches the working position;
- test 2 is the test with $\rho_F = 0.33$ and the person extending the arm after the robot reaches the working position;
- test 3 is the test with $\rho_F = 0.40$ and the person extending the arm before the robot reaches the working position;

- test 4 is the test with $\rho_F = 0.40$ and the person extending the arm after the robot reaches the working position;

The results of Test 1 are depicted in figures 5.40 and 5.41. These figures present frames from the test, both with data processed in MATLAB, indicated by the subscript m, and with photos acquired during the test, indicated by the subscript t. In the

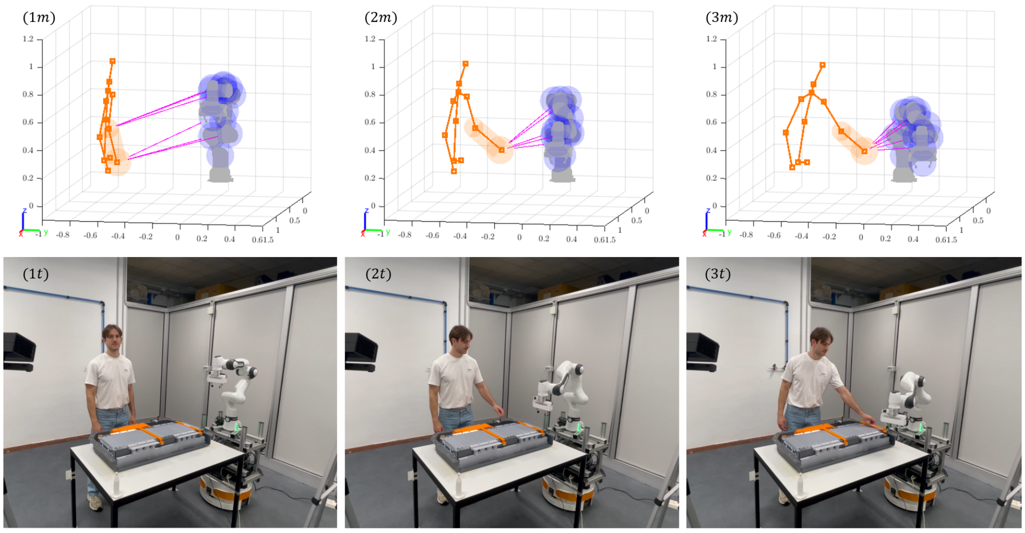


Fig. 5.40 Test 1: Frames of the test, part 1.

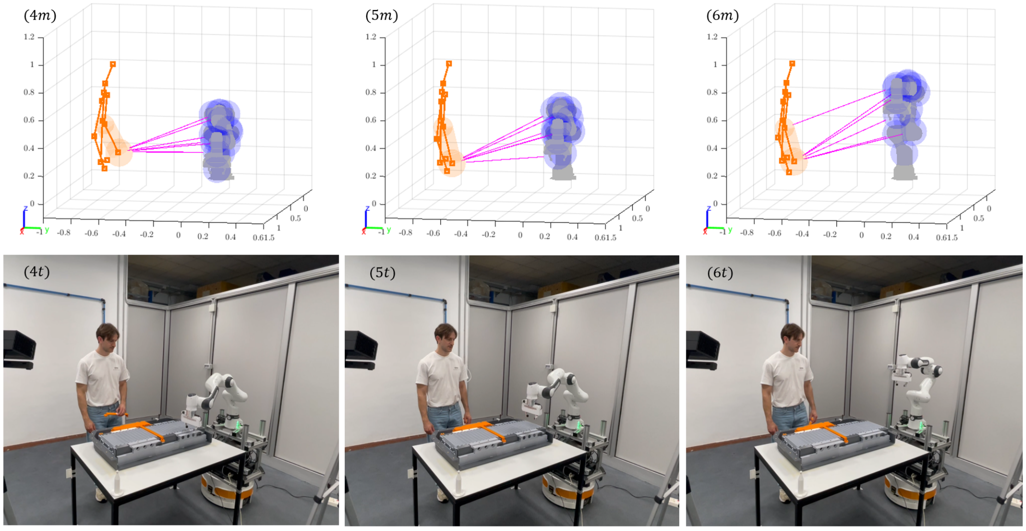


Fig. 5.41 Test 1: Frames of the test, part 2.

m images, the robot is shown with the collision avoidance spheres, the minimum distances highlighted in magenta, and the human operator's skeleton. Only the capsule constructed on the operator's left arm is shown to make the images more readable. In frame 1, the beginning of the test is shown. In frame 2, the operator starts moving his left arm towards the robot before it reaches the working position. In frame 3, the arm has breached the safety distance and the robot has deviated from its original path, avoiding collision. In frame 4, the operator has withdrawn their arm, and the robot has returned to its working position. In frame 5, the robot positions the EE in the lifting position, and in frame 6, it returns to its home position.

Test 2 presents characteristics similar to Test 1, except for the timing of when the operator moves their arm. In this test, the operator extends their arm after the robot has already reached the working position. This scenario tests the robot's ability to react to sudden movements in its workspace. In this test, the operator extends their arm towards the robot after the robot has reached its working position. The robot detects the arm and initiates the collision avoidance maneuver by generating a repulsive velocity that moves the EE away from its position, ensuring it maintains a safe distance from the operator. The robot then returns to its task position once the operator's arm is no longer in the critical proximity zone.
In figure 5.42, the paths of the EE during the collision avoidance maneuvers are shown for both Test 1 and Test 2. On these paths, the minimum distances between the EE and the human operator are indicated, as well as the corresponding values of the repulsive velocities.

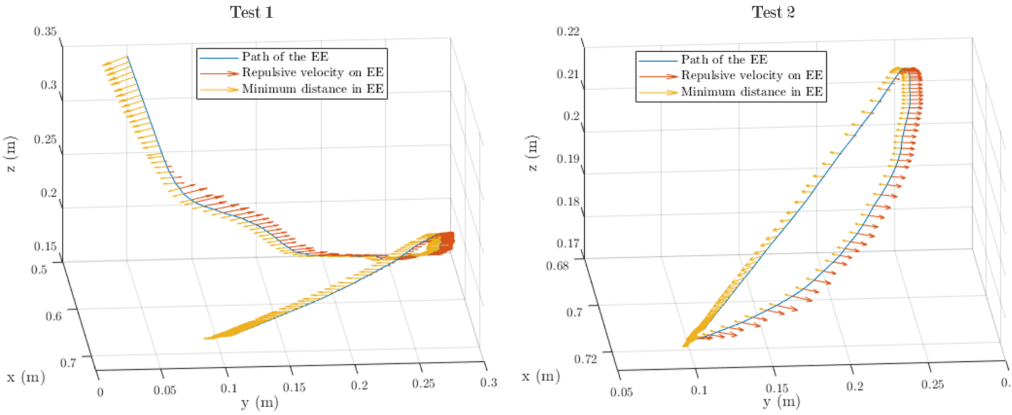


Fig. 5.42 Paths of the end effector during collision avoidance in Test 1 and Test 2, with indicated minimum distances and corresponding repulsive velocities.

The results of Test 4 are shown in figure 5.43 and 5.44, with $\rho_F = 0.40$ and the operator extending his arm after the robot has reached its working position. As for

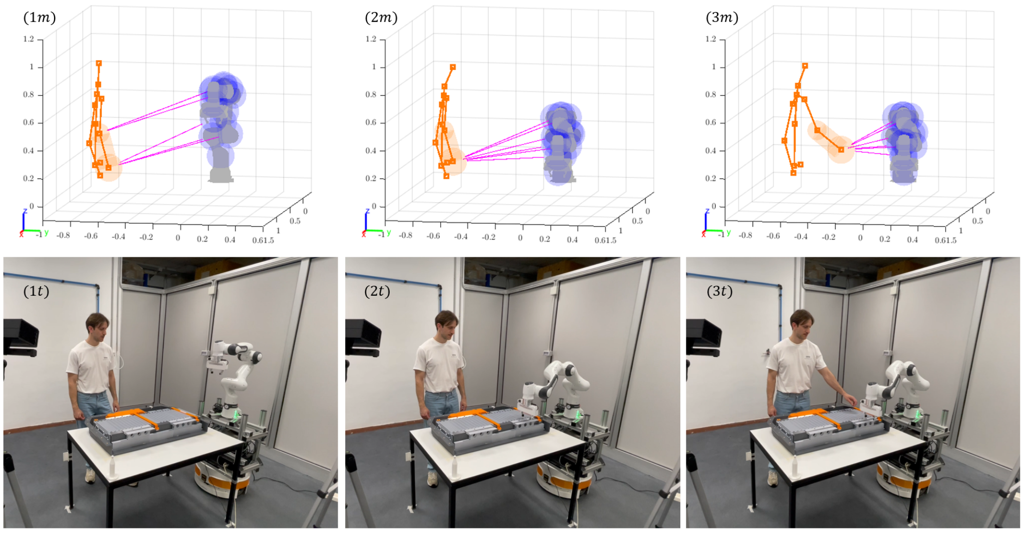


Fig. 5.43 Test 4: Frames of the test, part 1.

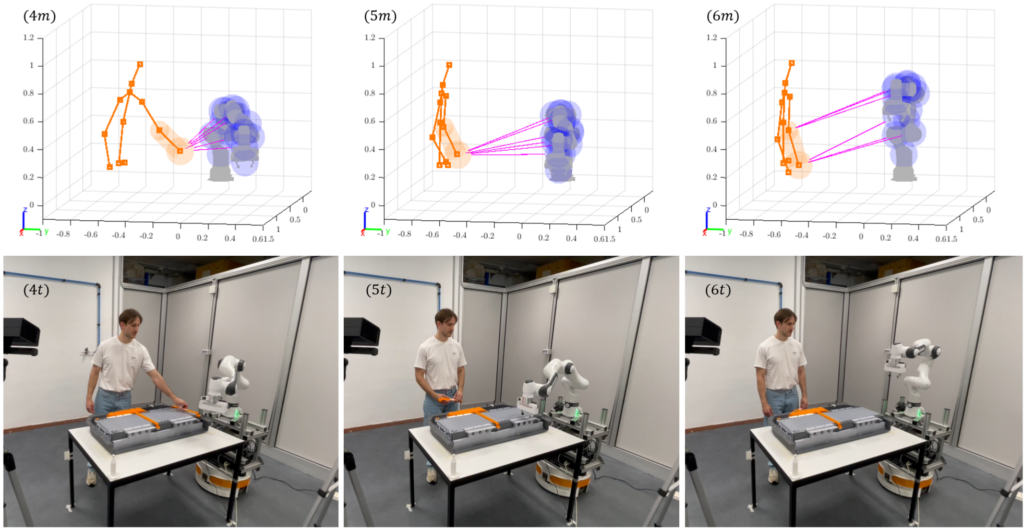


Fig. 5.44 Test 4: Frames of the test, part 2.

test 1, both the data processed in MATLAB, *m*, and the pictures taken during the test, *e*, are shown. In frame 1, the beginning of the test is shown. In frame 2, the robot reaches the working position. In frame 3, the operator moves his left arm towards the robot, which begins to move away from the person. In frame 4, the operator has his arm extended to grab the busbar, and the robot has successfully avoided the arm. In frame 5, the robot has returned to its working position to complete its task, and in frame 6, it returns to its home position.

It is particularly interesting to note that with the increase in the value of ρ_F, corresponding to an increase in the safety distance, the robot reacts more promptly to the human operator's movements. This is especially evident when comparing frame 2 of the first test, figure 5.40, and frame 3 of the fourth test, figure 5.43. In both frames, the position of the operator's left arm is approximately the same. However, in the first test, the robot remains stationary in its working position, while in the fourth test, the robot has already started to move away, demonstrating a quicker response to the human operator's movement.

In Figure 5.45, the paths of the EE during the collision avoidance maneuvers are shown for both Test 3 and Test 4, with indicated both the minimum distance of the EE from the operator and the repulsive velocities acting on EE. The movement with

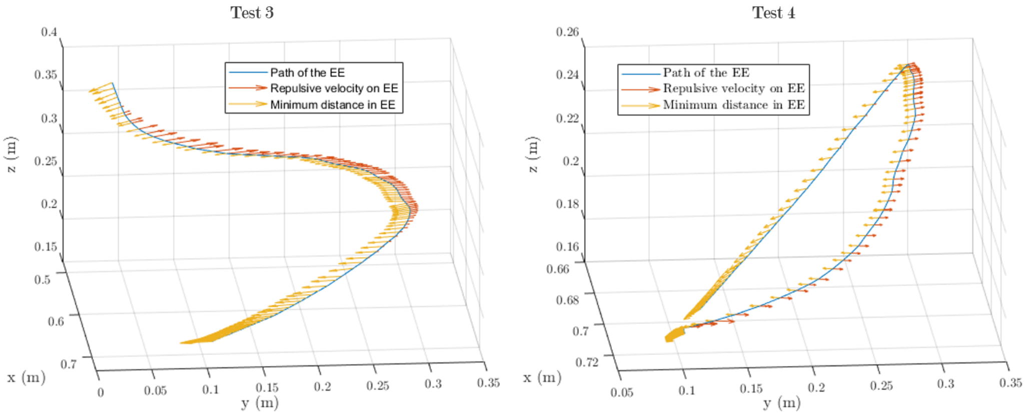


Fig. 5.45 Paths of the end effector during collision avoidance in Test 3 and Test 4, with indicated minimum distances and corresponding repulsive velocities.

$\rho_F = 0.40$ is smoother, as seen when comparing the EE trajectories in Test 1, Figure 5.42, and Test 3, Figure 5.45. However, this does not imply that choosing $\rho_F = 0.33$ is improper. A lower value of ρ_F allows the robot to continue its task for a longer period before being interrupted or potentially avoid interruption if the human does not breach the defined threshold. The choice of ρ_F thus depends on the specific working conditions, including the speed of the human operator's movements. In environments where the operator moves more slowly, a lower ρ_F can be appropriate, and vice versa.

In conclusion, all the tested conditions yielded satisfactory results, demonstrating the robustness and adaptability of the collision avoidance algorithm in the case proposed.

5.5 Conclusions

The experimental results from the collaborative robotic cell setup demonstrate the feasibility and effectiveness of integrating a mobile base with an anthropomorphic robotic arm for precise and safe operations in a shared workspace. The tests covered critical aspects of human-robot interaction, collision avoidance, and precise localization.

The collision avoidance tests for the Tiago mobile base showed that the robot could effectively navigate around a human operator, maintaining a safe distance and ensuring smooth operation. The algorithm ability to make the robot movement direction predictable can enhance the operator's comfort and confidence, demonstrating its potential in industrial contexts.

The localization tests confirmed that the robot could accurately position itself near the workbench to perform detailed tasks. Using Aruco markers and the Realsense camera provided precise positional data. The collision avoidance tests between the Franka robotic arm and the human operator demonstrated the robot ability to safely interact in close proximity with humans and in different conditions.

The development and use of a simulation tool were crucial in testing and refining the algorithms before real-world deployment. The simulation environment allowed for testing the control algorithms, before experimental tests in real world.

In summary, the chapter provides insights into developing and validating a collaborative robotic cell capable of safe human-robot interaction. The results highlight the potential for deploying such systems in industrial environments, balancing safety and efficiency.

Chapter 6

Conclusions and future works

In this work, the integration of vision-based algorithms in a collaborative robotic workcell for the disassembly of electric and electronic waste, with a special focus on vehicle batteries (EVBs) was explored. The primary objective was to enhance the disassembly process by employing robots to work alongside human operators, thereby improving efficiency and safety. The study approach involved several key steps, from detailed case studies and dynamic modeling to the development and testing of collaborative robotics algorithms.

The dissertation began by examining the role of robotics in sustainable waste management within the circular economy framework. Emphasis was placed on the necessity for advanced technological solutions to address resource scarcity and environmental sustainability. Robotics and AI were highlighted as pivotal in enabling precise and efficient waste sorting and processing. A detailed case study of about the disassembly of an EVB was examined to identify specific disassembly tasks and determine the optimal robotic solution.

This analysis led to selecting a robotic manipulator equipped with vision-based control systems as the most suitable for the disassembly process, focusing on improved localization and ensuring operator safety.

Dynamic model of the robotic system were developed to monitor interactions between the robots and their movements within the workspace. A UR5 and a MiR250 were used for the definition of the model. The model was identified and validated using Optitrack body tracking system. Particular attention was given to the mobile robot, focusing on the forces exchanged between the wheels and the floor.

The development and implementation of collaborative robotics algorithms were central to this work. This included gaze tracking for robot control and collision avoidance algorithms. Gaze tracking algorithms were studied for its potential in collaborative robotics applications, and tested both in simulation and with a UR5 robot. An algorithm that allows to avoid collision in a predictable way was implemented and tested with mobile robot, using a Turtlebot, both in simulation and experiments, with fixed obstacle.

Practical implementation of the collaborative robotic cell for EVB disassembly involved integrating vision systems, precise localization techniques, and effective collision avoidance algorithms. The optimal robotic solutions was determined. The disassembly analysis led to the selection of a Tiago Base mobile robot and a Franka FR3 robotic arm as the most suitable solutions for the disassembly process. Two Kinect V2 cameras were chosen for tracking the human operator's skeleton, and a Realsense D435 camera were selected for reading Aruco markers. The Realsense was positioned on the robotic system to ensure accurate spatial localization. A mocked EVB was 3D-printed for tests.
Experimental setups and tests were conducted to validate system performance in three key areas: collision avoidance between the mobile robot and the human operator, precise positioning of the robotic arm relative to the workbench, and collision avoidance between the robotic arm and the human operator. The tests demonstrated the system potential for performing complex disassembly tasks safely and efficiently. The results confirmed that the combination of advanced vision-based control and collaborative robotics can enhances the efficiency and safety of EVB disassembly processes.
The final phase of the research focused on the real-world application of the developed algorithms. The mobile robot path was optimized to avoid collisions with human operators predictably, and the anthropomorphic robotic arm was tested for its ability to continue tasks without interruption despite human interference. The localization algorithm effectiveness was verified by the robot capability to position itself accurately and perform precise tasks.
The developed methodologies and technologies have broader implications for industrial automation and waste management.

The ongoing development of the collaborative robotic cell will focus on several key enhancements to improve efficiency, safety, and adaptability in industrial contexts. One of the primary future directions involves transferring and adapting the dynamic model of the UR5-MiR250 to the Tiago-Franka robotic solution. This model will be identified and validated using the OptiTrack system, facilitating the testing of collaborative algorithms in a comprehensive dynamic environment that integrates the model with the simulation tool.

One promising direction is to upgrade the current Kinect V2 cameras with new Orbbec Femto Mega cameras, which integrate NVIDIA Jetson Nano boards, allowing a single computer to handle both visual processing and robot control, which can simplify the system architecture. They also have the possibility to use a synchronization hub that up to eight cameras, enabling comprehensive tracking of the operator's skeleton throughout the entire workcell, thus enhancing coverage and precision.

The tests on the interaction between Tiago robot and operator indicate the potential for developing a more intuitive interaction model between humans and robots, suggesting to define a code of conduct among them. The Realsense camera mounted on the Tiago robot could also be integrated with skeleton tracking capabilities, enabling the robot to avoid collisions not only within the collaborative cell, but also during its motion in the industrial environment.

To further improve the capabilities of the anthropomorphic robot, adding an extra camera on the end effector could significantly enhance the robot positioning accuracy for tasks requiring high precision.

Additionally, integrating gaze tracking presents a promising avenue for enhancing human-robot interaction. By using gaze to indicate robots their working areas or to advise them of personal intentions, operators could significantly improve the efficiency and fluidity of collaboration with robots. Such advancements in gaze tracking could play a crucial role in developing more intuitive and seamless interactions within collaborative robotics, thereby contributing to safer and more efficient industrial practices.

These developments promise to extend the current work towards the management of a group of robots collaborating with a team of operators.

The methodologies and technologies developed through this research have the potential for broader application across various domains of industrial automation and waste management, further extending their impact and utility.

References

[1] Martin Geissdoerfer, Paulo Savaget, Nancy Bocken, and Erik Hultink. The circular economy – a new sustainability paradigm? *Journal of Cleaner Production*, 143:757–768, 02 2017.

[2] Julian Kirchherr, Denise Reike, and M.P. Hekkert. Conceptualizing the circular economy: An analysis of 114 definitions. *SSRN Electronic Journal*, 127, 09 2017.

[3] Marina Pieroni, Tim McAloone, and Daniela Pigosso. Business model innovation for circular economy: Integrating literature and practice into a conceptual process model. *Proceedings of the Design Society: International Conference on Engineering Design*, 1:2517–2526, 07 2019.

[4] Shubhangini Rajput and Surya Prakash Singh. Connecting circular economy and industry 4.0. *International Journal of Information Management*, 49:98–113, 2019.

[5] Steffen Blömeke, Julian Rickert, Mark Mennenga, Sebastian Thiede, Thomas Spengler, and Christoph Herrmann. Recycling 4.0 – mapping smart manufacturing solutions to remanufacturing and recycling operations. *Procedia CIRP*, 90:600–605, 01 2020.

[6] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K.E. Lorber, and R. Pomberger. Digitalisation and intelligent robotics in value chain of circular economy oriented waste management – a review. *Waste Management*, 95:476–492, 2019.

[7] Sathish Paulraj Gundupalli, Subrata Hait, and Atul Thakur. A review on automated sorting of source-separated municipal solid waste for recycling. *Waste management*, 60:56–74, 2017.

[8] Alexander Abad. Automated waste sorter with mobile robot delivery waste system. 2013.

[9] Antonio Luca Alfeo, Eduardo Castelló Ferrer, Yago Lizarribar Carrillo, Arnaud Grignard, Luis Alonso Pastor, Dylan T Sleeper, Mario GCA Cimino, Bruno Lepri, Gigliola Vaglini, Kent Larson, et al. Urban swarms: A new approach for autonomous waste management. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4233–4240. IEEE, 2019.

[10] Ralph Sherwin A Corpuz and John Clifford R Orquiza. Utilization of fuzzy logic control in a waste robot. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pages 1–6. IEEE, 2018.

[11] Esther Álvarez-de-los Mozos, Arantxa Rentería-Bilbao, and Fernando Díaz-Martín. Weee recycling and circular economy assisted by collaborative robots. *Applied Sciences*, 10(14):4800, 2020.

[12] European Union Law. Directive 2012/19/eu of the european parliament and of the council of 4 july 2012 on waste electrical and electronic equipment (weee) (recast) (text with eea relevance). https://eur-lex.europa.eu/eli/dir/2012/19/2018-07-04. Accessed: 26/02/2024.

[13] European Union Law. Directive 2011/65/eu of the european parliament and of the council of 8 june 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (recast) (text with eea relevance). https://eur-lex.europa.eu/eli/dir/2011/65/oj. Accessed: 26/02/2024.

[14] Esther Alvarez-de-los Mozos and Arantxa Renteria. Collaborative robots in e-waste management. *Procedia Manufacturing*, 11:55–62, 2017.

[15] Tsai Chi Kuo. The construction of a collaborative-design platform to support waste electrical and electronic equipment recycling. *Robotics and Computer-Integrated Manufacturing*, 26(1):100–108, 2010.

[16] Hendrik Poschmann, Holger Brueggemann, and Daniel Goldmann. Disassembly 4.0: A review on using robotics in disassembly tasks as a way of automation. *Chemie Ingenieur Technik*, 92(4):341–359, 2020.

[17] Mohammad Alshibli, Ahmed El Sayed, Ozden Tozanli, Elif Kongar, Tarek M Sobh, and Surendra M Gupta. A decision maker-centered end-of-life product recovery system for robot task sequencing. *Journal of Intelligent & Robotic Systems*, 91:603–616, 2018.

[18] J Li, M Barwood, and S Rahimifard. A multi-criteria assessment of robotic disassembly to support recycling and recovery. *Resources, Conservation and Recycling*, 140:158–165, 2019.

[19] Jie Li, Michael Barwood, and Shahin Rahimifard. An automated approach for disassembly and recycling of electric vehicle components. In *2014 IEEE International Electric Vehicle Conference (IEVC)*, pages 1–6. IEEE, 2014.

[20] Jie Li, Michael Barwood, and Shahin Rahimifard. Robotic disassembly for increased recovery of strategically important materials from electrical vehicles. *Robotics and Computer-Integrated Manufacturing*, 50:203–212, 2018.

[21] Marco Marconi, Giacomo Palmieri, Massimo Callegari, and Michele Germani. Feasibility study and design of an automatic system for electronic components disassembly. *Journal of Manufacturing Science and Engineering*, 141(2):021011, 2019.

[22] Piotr Nowakowski and Teresa Pamuła. Application of deep learning object classifier to improve e-waste collection planning. *Waste Management*, 109:1–9, 2020.

[23] Callie W Babbitt, Hema Madaka, Shahana Althaf, Barbara Kasulaitis, and Erinn G Ryen. Disassembly-based bill of materials data for consumer electronic products. *Scientific Data*, 7(1):251, 2020.

[24] S Ruggeri, G Fontana, V Basile, M Valori, and Irene Fassi. Micro-robotic handling solutions for pcb (re-) manufacturing. *Procedia Manufacturing*, 11:441–448, 2017.

[25] Ioannis Chatzikonstantinou, Dimitrios Giakoumis, and Dimitrios Tzovaras. A new shopfloor orchestration approach for collaborative human-robot device disassembly. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 225–230. IEEE, 2019.

[26] Kai Li, Quan Liu, Wenjun Xu, Jiayi Liu, Zude Zhou, and Hao Feng. Sequence planning considering human fatigue for human-robot collaboration in disassembly. *Procedia CIRP*, 83:95–104, 2019.

[27] Arantxa Renteria and Esther Alvarez-de-los Mozos. Human-robot collaboration as a new paradigm in circular economy for weee management. *Procedia Manufacturing*, 38:375–382, 2019.

[28] Tobias Elwert, Daniel Goldmann, Felix Römer, Matthias Buchert, Cornelia Merz, Doris Schueler, and Juergen Sutter. Current developments and challenges in the recycling of key components of (hybrid) electric vehicles. *Recycling*, 1(1):25–60, 2015.

[29] Evan M Leon and Shelie A Miller. An applied analysis of the recyclability of electric vehicle battery packs. *Resources, Conservation and Recycling*, 157:104593, 2020.

[30] Carlos M Costa, João C Barbosa, Renato Gonçalves, Helder Castro, FJ Del Campo, and Senentxu Lanceros-Méndez. Recycling and environmental issues of lithium-ion batteries: Advances, challenges and opportunities. *Energy Storage Materials*, 37:433–465, 2021.

[31] Alexandre Beaudet, François Larouche, Kamyab Amouzegar, Patrick Bouchard, and Karim Zaghib. Key challenges and opportunities for recycling electric vehicle battery materials. *Sustainability*, 12(14):5837, 2020.

[32] Thomas P Hendrickson, Olga Kavvada, Nihar Shah, Roger Sathre, and Corinne D Scown. Life-cycle implications and supply chain logistics of electric vehicle battery recycling in california. *Environmental Research Letters*, 10(1):014011, 2015.

[33] European Union Law. Regulation (eu) 2023/1542 of the european parliament and of the council of 12 july 2023 concerning batteries and waste batteries, amending directive 2008/98/ec and regulation (eu) 2019/1020 and repealing directive 2006/66/ec (text with eea relevance). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32023R1542&qid=1710708373207. Accessed: 17/03/2024.

[34] Quentin Hoarau and Etienne Lorang. An assessment of the european regulation on battery recycling for electric vehicles. *Energy Policy*, 162:112770, 2022.

[35] Jens Schäfer, Ramona Singer, Janna Hofmann, and Jürgen Fleischer. Challenges and solutions of automated disassembly and condition-based remanufacturing of lithium-ion battery modules for a circular economy. *Procedia Manufacturing*, 43:614–619, 2020.

[36] Monsuru Olalekan Ramoni and Hong-Chao Zhang. End-of-life (eol) issues and options for electric vehicle batteries. *Clean Technologies and Environmental Policy*, 15:881–891, 2013.

[37] Lluc Canals Casals, Beatriz Amante García, and Lázaro V Cremades. Electric vehicle battery reuse: Preparing for a second life. *Journal of Industrial Engineering and Management*, 10(2):266–285, 2017.

[38] Chris White and Lukas G Swan. Pack-level performance of electric vehicle batteries in second-life electricity grid energy services. *Journal of Energy Storage*, 57:106265, 2023.

[39] Simon Glöser-Chahoud, Sandra Huster, Sonja Rosenberg, Sabri Baazouzi, Steffen Kiemel, Soumya Singh, Christian Schneider, Max Weeber, Robert Miehe, and Frank Schultmann. Industrial disassembling as a key enabler of circular economy solutions for obsolete electric vehicle battery systems. *Resources, Conservation and Recycling*, 174:105735, 2021.

[40] Bin Huang, Zhefei Pan, Xiangyu Su, and Liang An. Recycling of lithium-ion batteries: Recent advances and perspectives. *Journal of Power Sources*, 399:274–286, 2018.

[41] Mengyuan Chen, Xiaotu Ma, Bin Chen, Renata Arsenault, Peter Karlson, Nakia Simon, and Yan Wang. Recycling end-of-life electric vehicle lithium-ion batteries. *Joule*, 3(11):2622–2646, 2019.

[42] Liu Yun, Duy Linh, Li Shui, Xiongbin Peng, Akhil Garg, My Loan Phung Le, Saeed Asghari, and Jayne Sandoval. Metallurgical and mechanical methods

for recycling of lithium-ion battery pack for electric vehicles. *Resources, Conservation and Recycling*, 136:198–208, 2018.

[43] Lluc Canals Casals and Beatriz Amante García. Assessing electric vehicles battery second life remanufacture and management. *Journal of Green Engineering*, 6(1):77–98, 2016.

[44] Gavin Harper, Roberto Sommerville, Emma Kendrick, Laura Driscoll, Peter Slater, Rustam Stolkin, Allan Walton, Paul Christensen, Oliver Heidrich, Simon Lambert, et al. Recycling lithium-ion batteries from electric vehicles. *nature*, 575(7781):75–86, 2019.

[45] Jan Schmitt, Hannes Haupt, Michael Kurrat, and Annika Raatz. Disassembly automation for lithium-ion battery systems using a flexible gripper. In *2011 15th International Conference on Advanced Robotics (ICAR)*, pages 291–297. IEEE, 2011.

[46] Jia Mao, Dou Hong, Zhe Chen, Ma Changhai, Li Weiwen, and Ju Wang. Disassembly sequence planning of waste auto parts. *Journal of the Air & Waste Management Association*, 71(5):607–619, 2021.

[47] Héctor Rallo, G Benveniste, I Gestoso, and B Amante. Economic analysis of the disassembling activities to the reuse of electric vehicles li-ion batteries. *Resources, conservation and recycling*, 159:104785, 2020.

[48] Qingdi Ke, Peng Zhang, Lei Zhang, and Shouxu Song. Electric vehicle battery disassembly sequence planning based on frame-subgroup structure combined with genetic algorithm. *Frontiers in Mechanical Engineering*, 6:576642, 2020.

[49] M Alfaro-Algaba and F Javier Ramirez. Techno-economic and environmental disassembly planning of lithium-ion electric vehicle battery packs for remanufacturing. *Resources, Conservation and Recycling*, 154:104461, 2020.

[50] Kai Meng, Guiyin Xu, Xianghui Peng, Kamal Youcef-Toumi, and Ju Li. Intelligent disassembly of electric-vehicle batteries: a forward-looking overview. *Resources, Conservation and Recycling*, 182:106207, 2022.

[51] Martin Choux, Eduard Marti Bigorra, and Ilya Tyapin. Task planner for robotic disassembly of electric vehicle battery pack. *Metals*, 11(3):387, 2021.

[52] Hendrik Poschmann, Holger Brüggemann, and Daniel Goldmann. Fostering end-of-life utilization by information-driven robotic disassembly. *Procedia CIRP*, 98:282–287, 2021.

[53] Hao Yin, Jinhua Xiao, and Guoxian Wang. Human-robot collaboration remanufacturing for uncertain disassembly in retired battery recycling. In *2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, pages 595–598. IEEE, 2022.

[54] Kathrin Wegener, Stefan Andrew, Annika Raatz, Klaus Dröder, and Christoph Herrmann. Disassembly of electric vehicle batteries using the example of the audi q5 hybrid system. *Procedia Cirp*, 23:155–160, 2014.

[55] Kathrin Wegener, Wei Hua Chen, Franz Dietrich, Klaus Dröder, and Sami Kara. Robot assisted disassembly for the recycling of electric vehicle batteries. *Procedia Cirp*, 29:716–721, 2015.

[56] Wei Jie Tan, Christina Mei Mei Chin, Akhil Garg, and Liang Gao. A hybrid disassembly framework for disassembly of electric vehicle batteries. *International Journal of Energy Research*, 45(5):8073–8082, 2021.

[57] Ian Kay, Siamak Farhad, Ajay Mahajan, Roja Esmaeeli, and Sayed Reza Hashemi. Robotic disassembly of electric vehicles' battery modules for recycling. *Energies*, 15(13):4856, 2022.

[58] Sebastian Blankemeyer, Denise Wiens, Tobias Wiese, Annika Raatz, and Sami Kara. Investigation of the potential for an automated disassembly process of bev batteries. *Procedia CIRP*, 98:559–564, 2021.

[59] International Federation of Robotics. Demystifying collaborative industrial robots, 2020. Accessed on April 2, 2024.

[60] Jan F Hellmuth, Nicholas M DiFilippo, and Musa K Jouaneh. Assessment of the automation potential of electric vehicle battery disassembly. *Journal of Manufacturing Systems*, 59:398–412, 2021.

[61] WeberAuto. Chevrolet bolt ev battery disassembly, 2018. https://www.youtube.com/watch?v=ssU2mjiNi_Q&ab_channel=WeberAuto [Accessed: 4 April 2024].

[62] Christoph Herrmann, Annika Raatz, Mark Mennenga, Jan Schmitt, and Stefan Andrew. Assessment of automation potentials for the disassembly of automotive lithium ion battery systems. In *Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of California at Berkeley, Berkeley, USA, May 23-25, 2012*, pages 149–154. Springer, 2012.

[63] Laura Lander, Chris Tagnon, Viet Nguyen-Tien, Emma Kendrick, Robert JR Elliott, Andrew P Abbott, Jacqueline S Edge, and Gregory J Offer. Breaking it down: A techno-economic assessment of the impact of battery pack design on disassembly costs. *Applied Energy*, 331:120437, 2023.

[64] Andrea Raviola, Andrea De Martin, Roberto Guida, Stefano Pastorelli, Stefano Mauro, and Massimo Sorli. Identification of a ur5 collaborative robot dynamic parameters. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 69–77. Springer, 2021.

[65] Mads Hvilshøj, Simon Bøgh, Oluf Skov Nielsen, and Ole Madsen. Autonomous industrial mobile manipulation (aimm): past, present and future. *Industrial Robot: An International Journal*, 39(2):120–135, 2012.

[66] Brad Hamner, Seth Koterba, Jane Shi, Reid Simmons, and Sanjiv Singh. An autonomous mobile manipulator for assembly tasks. *Autonomous Robots*, 28:131–149, 2010.

[67] Qiang Huang, Shigeki Sugano, and Kazuo Tanie. Motion planning for a mobile manipulator considering stability and task constraints. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 3, pages 2192–2198. IEEE, 1998.

[68] Jiewu Leng, Dewen Wang, Weiming Shen, Xinyu Li, Qiang Liu, and Xin Chen. Digital twins-based smart manufacturing system design in industry 4.0: A review. *Journal of manufacturing systems*, 60:119–137, 2021.

[69] CK Lo, Chun-Hsien Chen, and Ray Y Zhong. A review of digital twin in product design and development. *Advanced Engineering Informatics*, 48:101297, 2021.

[70] Niki Kousi, Christos Gkournelos, Sotiris Aivaliotis, Christos Giannoulis, George Michalos, and Sotiris Makris. Digital twin for adaptation of robots' behavior in flexible robotic assembly lines. *Procedia manufacturing*, 28:121–126, 2019.

[71] Mir-mobile industrial robots. Accessed on May 5, 2024.

[72] Siciliano Bruno, Sciavicco Lorenzo, Villani Luigi, and Oriolo Giuseppe. Robotics: modelling, planning and control, 2010. *Cited on*, 1, 2010.

[73] Ur-universal robots. Accessed on May 5, 2024.

[74] Ur-universal robots denavit hartenberg data. Accessed on May 5, 2024.

[75] Hans Pacejka. *Tire and vehicle dynamics*. Elsevier, 2005.

[76] Jiahui Liu, Jiannan Chi, Huijie Yang, and Xucheng Yin. In the eye of the beholder: A survey of gaze tracking techniques. *Pattern Recognition*, 132:108944, 2022.

[77] Vaibhav V Unhelkar, Claudia Pérez-D'Arpino, Leia Stirling, and Julie A Shah. Human-robot co-navigation using anticipatory indicators of human walking motion. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6183–6190. IEEE, 2015.

[78] Blake Holman, Abrar Anwar, Akash Singh, Mauricio Tec, Justin Hart, and Peter Stone. Watch where you're going! gaze and head orientation as predictors for social robot navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3553–3559. IEEE, 2021.

[79] Valerio Belcamino, Miwa Takase, Mariya Kilina, Alessandro Carfì, Akira Shimada, Sota Shimizu, and Fulvio Mastrogiovanni. Gaze-based intention recognition for human-robot collaboration. *arXiv preprint arXiv:2405.07570*, 2024.

[80] Guangtao Zhang, John Paulin Hansen, and Katsumi Minakata. Hand-and gaze-control of telepresence robots. In *Proceedings of the 11th acm symposium on eye tracking research & applications*, pages 1–8, 2019.

[81] Photchara Ratsamee, Yasushi Mae, Kenichi Ohara, Tomohito Takubo, and Tatsuo Arai. Human–robot collision avoidance using a modified social force model with body pose and face orientation. *International Journal of Humanoid Robotics*, 10(01):1350008, 2013.

[82] Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding*, 98(1):4–24, 2005.

[83] Ibrahim Shehi Shehu, Yafei Wang, Athuman Mohamed Athuman, and Xianping Fu. Remote eye gaze tracking research: a comparative evaluation on past and recent progress. *Electronics*, 10(24):3165, 2021.

[84] Hiroyuki Sogo. Gazeparser: an open-source and multiplatform library for low-cost eye tracking and analysis. *Behavior research methods*, 45:684–695, 2013.

[85] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the symposium on eye tracking research and applications*, pages 207–210, 2014.

[86] Opengazer: open-source gaze tracker for ordinary webcams. Accessed on May 25, 2024.

[87] openeyes – eye tracking systems used for casino gaming, paysafecard payments and more. Accessed on May 25, 2024.

[88] Itu gaze tracker - gaze tracking library. Accessed on May 25, 2024.

[89] antoinelame - gazetracking. Accessed on May 25, 2024.

[90] Baltrusaitis Tadas, Zadeh Amir, Lim Yao Chong, and Louis-Morency Philippe. Openface 2.0: Facial behavior analysis toolkit. In *13th IEEE International Conference on Automatic Face & Gesture Recognition*, 2018.

[91] Openface 2.2.0: a facial behavior analysis toolkit. Accessed on May 25, 2024.

[92] Eyeware gazesense sdk. Accessed on May 25, 2024.

[93] Tobii. Accessed on May 25, 2024.

[94] Tobii pro glasses 3. Accessed on May 25, 2024.

[95] Tobii pro sparks. Accessed on May 25, 2024.

[96] Talabattula Sai Abhishek, Daniel Schilberg, and Arockia Selvakumar Arockia Doss. Retracted: Obstacle avoidance algorithms: A review. In *IOP Conference Series: Materials Science and Engineering*, volume 1012, page 012052. IOP Publishing, 2021.

[97] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[98] Richard Volpe and Pradeep Khosla. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1423–1436, 1990.

[99] Elon Rimon. *Exact robot navigation using artificial potential functions*. Yale University, 1990.

[100] Christopher I Connolly and Roderic A Grupen. The applications of harmonic functions to robotics. *Journal of robotic Systems*, 10(7):931–946, 1993.

[101] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[102] Animesh Chakravarthy and Debasish Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(5):562–574, 1998.

[103] Sami Haddadin, Holger Urbanek, Sven Parusel, Darius Burschka, Jürgen Roßmann, Alin Albu-Schäffer, and Gerd Hirzinger. Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3109–3116. IEEE, 2010.

[104] Jürgen Guldner and Vadim I Utkin. Sliding mode control for gradient tracking and robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation*, 11(2):247–254, 1995.

[105] Robin R Murphy. *Introduction to AI robotics*. MIT press, 2019.

[106] Zhang Long. Virtual target point-based obstacle-avoidance method for manipulator systems in a cluttered environment. *Engineering Optimization*, 52(11):1957–1973, 2020.

[107] Omur Arslan and Daniel E Koditschek. Sensor-based reactive navigation in unknown convex sphere worlds. *The International Journal of Robotics Research*, 38(2-3):196–223, 2019.

[108] Randal W Beard and Timothy W McLain. Motion planning using potential fields. 2003.

[109] Fabrizio Flacco, Torsten Kröger, Alessandro De Luca, and Oussama Khatib. A depth space approach to human-robot collision avoidance. In *2012 IEEE international conference on robotics and automation*, pages 338–345. IEEE, 2012.

[110] Matteo Parigi Polverini, Andrea Maria Zanchettin, and Paolo Rocco. A computationally efficient safety assessment for collaborative robotics applications. *Robotics and Computer-Integrated Manufacturing*, 46:25–37, 2017.

[111] Mohammad Safeea, Pedro Neto, and Richard Bearee. On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. *Robotics and Autonomous Systems*, 119:278–288, 2019.

[112] Yiwei Wang, Yixuan Sheng, Ji Wang, and Wenlong Zhang. Optimal collision-free robot trajectory generation based on time series prediction of human motion. *IEEE Robotics and Automation Letters*, 3(1):226–233, 2017.

[113] Kun Wei and Bingyin Ren. A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved rrt algorithm. *Sensors*, 18(2):571, 2018.

[114] Lingqi Zeng and Gary M Bone. Mobile robot collision avoidance in human environments. *International Journal of Advanced Robotic Systems*, 10(1):41, 2013.

[115] Kun Qian, Xudong Ma, Xianzhong Dai, and Fang Fang. Socially acceptable pre-collision safety strategies for human-compliant navigation of service robots. *Advanced Robotics*, 24(13):1813–1840, 2010.

[116] Daniel Carton, Wiktor Olszowy, and Dirk Wollherr. Measuring the effectiveness of readability for mobile robot locomotion. *International Journal of Social Robotics*, 8:721–741, 2016.

[117] Markus Koppenborg, Peter Nickel, Birgit Naber, Andy Lungfiel, and Michael Huelke. Effects of movement speed and predictability in human–robot collaboration. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 27(4):197–209, 2017.

[118] Matteo Melchiorre, Leonardo Sabatino Scimmi, Laura Salamina, Stefano Mauro, Stefano Pastorelli, et al. Robot collision avoidance based on artificial potential field with local attractors. In *ICINCO*, pages 340–350, 2022.

[119] Ur5 cb-series manual. Accessed on June 2, 2024.

[120] Ur5 cb-series datasheet. Accessed on June 2, 2024.

[121] Orbbec astra webpage. Accessed on June 2, 2024.

[122] Onrobot rg2 gripper webpage. Accessed on June 2, 2024.

[123] Turtlebot3 webpage. Accessed on June 2, 2024.

[124] Realsense d435 webpage. Accessed on June 7, 2024.

[125] Opencv - aruco marker webpage. Accessed on June 7, 2024.

[126] Microsoft kinect v2 webpage. Accessed on June 7, 2024.

[127] Tiago base webpage. Accessed on June 2, 2024.

[128] Franka robot webpage. Accessed on June 2, 2024.

[129] Franka robot interface documentation.

[130] Item webpage. Accessed on June 9, 2024.

[131] Matteo Melchiorre, Leonardo Sabatino Scimmi, Stefano Mauro, and Stefano Pastorelli. A novel constrained trajectory planner for safe human-robot collaboration. In *ICINCO*, pages 539–548, 2022.

[132] Jing Ren, Kenneth A McIsaac, Rajni V Patel, and Terry M Peters. A potential field model using generalized sigmoid functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):477–484, 2007.

[133] Christina Lichtenthäler, Tamara Lorenzy, and Alexandra Kirsch. Influence of legibility on perceived safety in a virtual human-robot path crossing task. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 676–681. IEEE, 2012.

[134] wikiros, gmapping. Accessed on June 19, 2024.

[135] wikiros, pmb-2 localization tutorial. Accessed on June 19, 2024.

[136] wikiros, amcl. Accessed on June 19, 2024.