STIIMA
Sistemi e Tecnologie Industriali Intelligenti
per il Manifatturiero Avanzato
Consiglio Nazionale delle Ricerche

National Research Council - Institute of Intelligent Industrial Systems
and Technologies for Advanced Manufacturing

DRIM

Ph.D. Program of National Interest in Robotics and Intelligent Machines

Administrative Headquarters: Università di Genova

Cooperative Interaction and Manipulation in Human-Robot and Multi-Robot Systems

by

Roberto Fausti

Thesis submitted for the degree of *Doctor of Philosophy* (38° cycle)
April 2026

Dr. Nicola Pedrocchi — Supervisor
Antonio Sgorbissa — Head of the PhD program

Thesis Jury:
Prof. Giovanni Legnani, *Università degli Studi di Brescia*
Prof. Ruggero Carli, *Università degli Studi di Padova*
Dr. Marco Faroni, *Politecnico di Milano*

Dibris

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Roberto Fausti
March 2026

Acknowledgements

First and foremost, I would like to thank Nicola, for its support and especially its patience throughout the research and the writing of this thesis. A special thanks also to Manuel, for his important help and invaluable tips. A thank you to Stefano, whose work on the mobile platforms has been fundamental. Thanks to all colleagues at CARI and SAR labs, from whom I have learned a lot and with whom I have shared many productive moments — and many unproductive ones. Thanks to the friends I met during my time in Barcelona, who turned my research period there into an experience that I will never forget.
Thanks to my family, who have supported me all these years and continue to support me to this day.

Abstract

With robotics transitioning from isolated, highly structured systems to open, dynamic environments, the ability to interact and cooperate with other intelligent agents (humans and robots) becomes a fundamental requirement. This thesis addresses cooperative interaction and manipulation along two complementary research directions: physical human–robot interaction and multi-robot formation control.

In the context of physical human–robot interaction, starting from the LuGre friction model, an elastoplastic admittance model is proposed that enables both elastic, reversible motion modifications and persistent offsets of the reference, allowing the operator to intuitively adapt the robot trajectory. The model is initially integrated into a variable-admittance control scheme compatible with non-redundant manipulators.

To address collaborative transportation with mobile manipulators, an alternative formulation of the elastoplastic model is presented, along with a whole-body control algorithm based on lexicographic optimization. The controller combines elastoplastic behavior with kinematic constraints and distributes the motion command between the manipulator and the mobile base, prioritizing the manipulator for elastic components and the base for plastic offsets. A trajectory-recovery mechanism, obtained by optimizing an intermediate reference within the lexicographic problem, enables controlled re-entry after a plastic adaptation. The approach is validated in a collaborative transportation task with a human operator.

Finally, in the multi-robot domain, a leader–follower control scheme is described in which the leader employs an expanded costmap to represent the environment from the formation perspective. The construction process ensures that the leader planned path is collision-free for the entire formation. A decentralized controller allows followers to maintain formation while enforcing a feasibility constraint that prevents the leader's velocity from exceeding the followers capabilities. The scheme is validated in simulation and on a real system through a collaborative transportation task involving a rigid object.

Table of contents

List of figures **vii**

List of tables **xi**

1 Physical Interaction **5**

1.1 Physical Human-Robot Interaction 5
1.2 Multi-robot systems 7
1.3 Impedance Control 9
1.4 Mobile Manipulators 12
1.5 Cooperative manipulation 15
1.6 Formation planning and control 17

2 Elastoplastic Controller for Interaction **21**

2.1 Introduction 22
2.1.1 Context 22
2.1.2 Contribution 23
2.1.3 Assumptions 24
2.2 Fundamentals of Physical Human-Robot Interaction Model 25
2.2.1 Admittance Control Formulation 25
2.2.2 Elasto-Plastic Formulation 27
2.3 Elasto-Plastic Model 30
2.3.1 LuGre's extension to model Elastic Displacement 30
2.3.2 Plastic Displacement Analysis 34
2.3.3 Model Extension for 3D Motion 35
2.4 Motion Reversal 38
2.4.1 Reset Condition 39
2.4.2 Complete Control Model 41

2.5 Experiments . . . 42
2.6 Summary . . . 46

3 Elastoplastic Control for Mobile Manipulation 47
3.1 Introduction . . . 49
3.1.1 Context . . . 49
3.1.2 Contribution . . . 50
3.1.3 Assumptions . . . 50
3.2 Fundamentals . . . 51
3.2.1 Mobile Manipulator . . . 51
3.2.2 Hierarchical Quadratic Programming . . . 52
3.3 Elastoplastic Formulation for Mobile Manipulation . . . 53
3.3.1 Elastic and Plastic Displacement . . . 55
3.3.2 Motion Reversal . . . 55
3.3.3 Reset Condition . . . 56
3.4 Optimization-based Elastoplastic Controller . . . 56
3.4.1 Arbitration Policy for Null-space Optimization . . . 57
3.5 Trajectory Recovery . . . 58
3.6 Design of Experiments . . . 61
3.6.1 Experimental Setup Description . . . 61
3.6.2 Experimental Setup Limits . . . 62
3.6.3 Assessment and Benchmarking . . . 63
3.7 Experiments and Results . . . 64
3.7.1 Assessment of Trajectory Tracking and Plastic Deformation . . . 64
3.7.2 Assessment of Trajectory Tracking and Recovery . . . 69
3.7.3 Benchmarking Algorithm Experiments . . . 69
3.7.4 Comparative Analysis . . . 70
3.7.5 Discussion . . . 74
3.8 Summary . . . 78

4 Control Shared Architecture for Cooperative Transport 79
4.1 Introduction . . . 81
4.1.1 Context . . . 81
4.1.2 Contributions and Assumptions . . . 82
4.2 Fundamentals . . . 83
4.2.1 Costmap-based Path Planning . . . 83

4.2.2 Formation control 85
4.3 Augmented Costmap 86
4.4 Leader-Follower Formation Control Architecture 89
4.4.1 Leader path tracking 89
4.4.2 Relative-pose regulation for rigid formation keeping 89
4.5 Compliance Control for Cooperative Manipulation 91
4.6 Experiments 93
4.6.1 Simulations 93
4.6.2 Real world experiments 95
4.6.3 Discussion 102
4.7 Summary 102

5 Conclusions **104**
5.1 Summary 104
5.2 Future Works 105

References **107**

List of figures

1 Examples of commercial mobile manipulators. From left to right: TIAgo from PAL Robotics, MoMa from Omron, Stretch from Boston Dynamics . 2

1.1 Standard application in industrial settings for physical Human-Robot Interaction. 6
1.2 Multi mobile robot cooperating in industrial settings (Credits to: Wikimedia Commons). 8
1.3 Cartesian Impedance Control in End-Effector Frame (Credits to: "Advanced Robotics for Manufacturing", by Emma Brandberg et al.) 10
1.4 Industrial Mobile Manipulator (Credits to: Robotnik Automation S.L.) . . . 13
1.5 Advanced cooperative manipulation of a saw (Credits to: Peternel L. [101]). 16
1.6 Collaborative Transportation of a soft material. 17
1.7 Formation control for transportation of components (Credits to: Alonso-Mora J., et al. [8]) 18

2.1 Control scheme considered. The elasto-plastic controller is implemented in Cartesian space and computes a deformation x_d from the actual trajectory. . 26
2.2 Representation of the LuGre friction model. 31
2.3 Plot of functions $\alpha(z)$ (blue line) and $\dot{\alpha}(z)$ (dashed red line). $z_{ba} = 0.5$, $z_{ss} = 1.5$. 32
2.4 Elastic response of LuGre model extended with stiction 33
2.5 Elasto-plastic behavior. Three force steps are applied, from 0 s to 2 s, from 4 s to 6 s, and from 8 s to 10 s, where the first and third have the same amplitude, a fifth of the second. $z_{ba} = 0.03$ m, $z_{ss} = 0.1$ m 36
2.6 Comparison between the state variables z (blue line) and w (dashed red line) without and with the introduction of r (dashed yellow line). 40

2.7 Collaborative cell. The 6-DoF UR10e robot includes a force-torque sensor on the wrist. The robot is mounted on a linear guide which, for the purpose of this work, was kept fixed. 43
2.8 Deformation from steady-state condition. (a) Forces; (b) state variables; (c) trajectory deformation 44
2.9 Deformation during trajectory. x-component (blue line), y-component (dashed red line), z-component (dotted yellow line). (a) Applied force; (b) z state variables; (c) deformation of the trajectory (x_d) computed by the controller . 45

3.1 Visual Structure of the HQP elastoplastic controller for mobile manipulators 54
3.2 Plastic deformation of a trajectory r 59
3.3 Recovery of the original trajectory. An intermediate Cartesian reference $x_d(t)$ is computed and moved such that the actual position of the end effector $f_k(q(t))$ returns on the precomputed path. 60
3.4 Tiago Pro used for the experiments, mounting Robotiq force-torque sensor and gripper 61
3.5 Graphical representation of the experiment 62
3.6 Snapshots of the experiment 65
3.7 Trajectory tracking and plastic deformation: Cartesian position of the end effector. Reference trajectory in green, computed Cartesian reference in orange, forward kinematics obtained from the control action values in blue, and forward kinematics of the actual robot position in red. 67
3.8 Trajectory tracking and plastic deformation: Filtered wrench provided to the elastoplastic controller. 67
3.9 Trajectory tracking and plastic deformation: norm of state variable vector **z**. Values of z_{ba} and z_{ss} are represented as dashed green line and solid red line, respectively. 68
3.10 Trajectory tracking and plastic deformation: Joint velocities. The top plot represents the velocities of the mobile base expressed in the world frame, while the remaining two plots represent the joint velocities of the manipulator. 68
3.11 Trajectory recovery: Cartesian position of the end effector. Reference trajectory in green, computed Cartesian reference in orange, forward kinematics obtained from the control action values in blue, and forward kinematics of the actual robot position in red 70
3.12 Trajectory recover: Filtered wrench provided to the elastoplastic controller . 71

3.13 Trajectory recover: norm of state variable vector **z**. Values of z_{ba} and z_{ss} are represented as dashed green line and solid red line, respectively. 71
3.14 Trajectory recovery: Joint velocities. The top plot shows the velocity of the mobile base expressed in the world frame, while the remaining two plots represent the joint velocities of the manipulator. 72
3.15 Benchmark Controller: Cartesian position of the end effector. Reference trajectory in green and forward kinematics of the actual robot position in red. 73
3.16 Benchmark Controller: Filtered wrench provided to the benchmark controller. 73
3.17 Effect of different values of σ_0 on the dynamics of z 77

4.1 Typical example of an industrial mobile manipulator with a lightweight collaborative robot mounted on a mobile platform. (Credits to Neobotix) . . 82
4.2 Team-based robotics system could support humans working in construction zones, manufacturing plants. (Credits to: Center for Robotics and Biosystems, Northwestern University) 83
4.3 Multilayer path and squeezing procedure 88
4.4 Control architecture. A reference velocity command is computed for the leader robot L. From the pose of the leader L and for every follower F_i, the relative transformation between the two is computed and fed to a PI controller plus feed-forward on the linear axes x, y and on the angular coordinate (yaw). 91
4.5 Navigation phases of the Gazebo simulation with a leader (left frame) and a follower (right frame): (a) A goal pose is set for the simulation, shown as green arrow; (b) The merged map is computed, only layer zero is shown (gray); (c) A Path is computed using A^*, shown in green; (d-f) The path is tracked by the robots. The inflated costmap, present in every figure, contains the objects on the map (black), the lethal cost area (cyan) and the exponentially decreasing inflated cost (red to blue). 96
4.6 Experimental setup: On the left, the Omron/Techman leader of the formation; on the right the Custom/UR10e follower 97
4.7 Trajectory followed by leader and follower mobile bases, with respect to global map frame. Bottom plot shows the Euclidian distance between the two bases. 99
4.8 Force measured at the follower end , w.r.t. end effector frame: x is aligned with the main direction of the formation and z is vertical downward 99
4.9 Snapshots of multi-robot cooperative transport experiments 101

4.10 Visualization of multi-robot cooperative transport using RVIz software. The most left panel shows the standard costmap computed by the leader robot. The remaining four panels show the execution of the test and the layer zero of the merged costmap 101

List of tables

2.1 Limitations of Impedance Control in Human-Guided Tasks 27
2.2 Comparison of Impedance and Elasto-Plastic Controllers in Human-Guided Tasks 30
2.3 Experiments parameters 42

3.1 Required LuGre model adjustments to obtain a model suitable for admittance control 54
3.2 Experiments parameters 66
3.3 Parameters Tuning Considerations 76

4.1 Map merging and path computing average time and standard deviation for ten repetitions on different map sizes, computed on a Jeston NX Orin directly on the leader AMR(MAP_j) and on a workstation(MAP_w). 94
4.2 Errors from nominal leader-follower constraint, measured in average distance and relative angle. Computed over ten repetitions for each environment. . . 95
4.3 Localization error from nominal leader-follower constraint, measured in average value and standard deviation on distance and relative angle, and costmap merging time over ten experiments 100

Introduction

From the early stages of robotics, robotic systems have been mainly designed for well-defined tasks executed in structured and controlled environments. The main objectives in these types of systems are typically formulated around precision, throughput, and repeatability. Interaction with external and autonomous agents, such as humans, is therefore either eliminated by design or tightly constrained through rigid protocols and physical separation using fenced workcells, dedicated safety zones, and preplanned trajectories, among other measures. In this classical regime, the robot can be treated as an isolated dynamical system, whose behavior is largely independent of other decision-making entities.

Contemporary robotics development is increasingly moving away from this paradigm for multiple factors. Technological advancements are a primary driver of this shift, as they have enabled robots to acquire robust capabilities in perception, computation, and control, thereby reducing the need for strict environmental structuring and preplanned tasks. Improvements in sensing and perception, enabled by reliable 3D vision, laser detection, and learning-based recognition, allow robots to estimate scene geometry despite variability in lighting and clutter. Meanwhile, modern control architectures fully utilize such information, providing an optimized plan that is both task- and kinematically appropriate to complete the assigned job safely.

Other essential driving forces in this transition come from the evolution of industry and society perception: Industry 4.0 and Industry 5.0 defined the pillars on which future industries and services are based, where robots are progressively expected to share workspaces with humans, to collaborate with other robots, and to operate in unstructured environments where uncertainty and change are intrinsic rather than exceptional. The definition of a new standard and regulation suitable for the expected transition amplifies the effect. Finally, societal and economic factors, such as demographic change, rising demand for healthcare support, and the need for flexible automation in manufacturing, have increased interest in robots that augment human capabilities rather than replace human labor. This results in real deployment environments that are increasingly dynamic, open, and only partially observable.

Figure 1 Examples of commercial mobile manipulators. From left to right: TIAgo from PAL Robotics, MoMa from Omron, Stretch from Boston Dynamics

As a consequence, new classes of collaborative, assistive, and service applications have become viable, going from manufacturing to co-work, warehouse logistics, domestic assistance, and clinical support. In these contexts, rather than being disturbances to reject, interactions between robots and other autonomous agents can be exploited, extending the possibilities offered by the automation system and introducing flexibility in the process.

An interesting robot architecture that is benefiting from this diffusion of new applications is the mobile manipulator. These robots, composed of a mobile base and a manipulator arm, combine the advantages of mobile robots, such as the wide workspace, with the dexterity and reachability of the manipulators, allowing them to complete tasks that a single mobile robot or manipulator is unable to achieve. Developed for the first time in the ’80s [17], the diffusion of mobile manipulators outside of the research field has been growing in recent years, also thanks to the proliferation of ready-to-use mobile manipulators on the consumer market. Some examples include TIAGo [116] from PAL Robotics, MoMa [96] from Omron, and Stretch [31] from Boston Dynamics.

Contributions

This thesis addresses robot physical interaction and cooperative manipulation through two complementary research threads. The first thread develops an elasto-plastic interaction control methodology for single robots, progressing from fixed-base manipulators to whole-body mobile manipulators, to combine reversible, elastic trajectory deformations for precision and stability with controlled, plastic trajectory offsets to support human-guided or contact-driven

motion adaptation. The second thread addresses cooperative transport with teams of mobile manipulators by proposing a formation-aware planning and control framework that enables safe navigation while maintaining rigid payload constraints, complemented by a compliance strategy to mitigate internal interaction loads during transport.

The main contributions of the thesis are as follows:

1. **Rheology-inspired elasto-plastic admittance controller for position-controlled manipulators.** Starting from the LuGre friction framework, the thesis develops an elasto-plastic model that can switch between reversible (elastic) and non-reversible (plastic) behavior as a function of the applied interaction force. It integrates it within an admittance control architecture suitable for stiff, position-controlled industrial robots.

2. **Plastic-state stabilization and motion-reversal handling in the elasto-plastic model.** To obtain physically consistent plastic behavior without undesired elastic restoring effects, the model is augmented with additional internal dynamics that suppress elastic contribution during plastic motion, and a reset mechanism is introduced to manage plastic-to-elastic transitions (including motion reversal) using a work-based condition.

3. **Whole-body elasto-plastic control for mobile manipulators via hierarchical quadratic programming.** The thesis extends elasto-plastic interaction control to mobile manipulators through a whole-body formulation based on hierarchical quadratic programming (HQP), enabling the robot to distribute motion and interaction consistently across base and arm. The approach explicitly supports both elastic trajectory deformation and plastic, permanent offset generation within a unified controller. A trajectory recovery mechanism is introduced to enable the system, when necessary, to converge back to the original nominal trajectory after a plastic offset has been introduced by promoting a recovery objective within the HQP stack and guiding an intermediate Cartesian reference accordingly.

4. **Leader-follower cooperative transport framework for teams of mobile manipulators.** For multi-robot cooperative transport, the thesis proposes an integrated architecture that combines centralized planning, based on an augmented costmap representation, with decentralized formation control, explicitly targeting the geometric and safety constraints induced by transporting a rigid payload.

5. **ROS-compatible open-source implementation** For each method presented in this thesis, an open-source implementation compatible with the popular ROS framework is provided

Thesis Structure

The dissertation is organized to progress from single-robot physical interaction control to whole-body mobile manipulation, and ultimately to cooperative transport with multiple mobile manipulators.

- **Chapter 1** provides the background needed for the remainder of the thesis. It reviews interaction control (impedance/admittance), summarizes key concepts in cooperative manipulation and shared control, and introduces mobile manipulators together with common control architectures.
- **Chapter 2** presents an elasto-plastic interaction model and its integration within an admittance control framework for position-controlled manipulators. The chapter discusses practical aspects such as 3D extension and motion reversal handling, and validates the approach experimentally on a fixed-base manipulator.
- **Chapter 3** extends the elasto-plastic interaction concept to mobile manipulators through a whole-body controller formulated as a hierarchical quadratic program (HQP). It details task prioritization, an arbitration strategy between base and arm behaviors, a trajectory recovery mechanism, and reports experimental results on a mobile manipulator in a collaborative transport scenario with a human partner.
- **Chapter 4** addresses multi-robot cooperative payload transportation as a formation-level problem. It proposes a leader-follower architecture combining centralized planning based on formation-aware mapping/costmap fusion with decentralized formation control. It incorporates a compliance strategy to mitigate internal interaction loads during transport. The chapter evaluates the approach using both simulation and real robotic platforms.
- **Chapter 5** concludes the dissertation by summarizing the main findings and outlining directions for future work.

Chapter 1

Physical Interaction

1.1 Physical Human-Robot Interaction

Human–Robot Interaction (HRI) is the research field that focuses on the study of system and control architectures that enable a robot to communicate and collaborate with human operators. The work in this field can be roughly divided into two main areas of research: social robotics and physical human-robot interaction. The former studies the communication aspect, primarily related to the interpretation of non-physical exchanges between the participants, such as human gestures or speech, and to provide appropriate responses in a social context.

Particularly relevant to this thesis is the latter: physical Human-Robot Interaction (pHRI) studies the physical aspect of interaction, related to the forces and motions exchanged during human-robot contact, and the expected response to enable intuitive and safe collaboration. pHRI addresses interaction scenarios in which humans and robots are mechanically coupled, either intermittently or continuously, and must adapt their behavior accordingly. Unlike conventional automation, these systems must operate under conditions of variability and dynamic human behavior, requiring particular attention to robot design and control.

The motivation for pHRI research lies not only in expanding the functional capabilities of robotic systems but also in ensuring that such systems can be deployed in human-centered environments without compromising safety and usability. In particular, due to the high number of risks to which the human user is subject when sharing the same space as an autonomous robot, several regulations have been approved to ensure safe applications and limit the possibility of serious accidents [1, 2].

The defining characteristic of pHRI is the presence of a closed physical loop between the human and the robot. In such systems, the human influences the robot's motion through applied forces or constraints, while the robot simultaneously affects the human through

Figure 1.1 Standard application in industrial settings for physical Human-Robot Interaction.

mechanical feedback. This bidirectional coupling introduces complex dynamics that depend on both the robot control architecture and the physical properties of the human user.

The scope of pHRI is broad, encompassing a wide variety of interaction scenarios. These include cooperative manipulation tasks, where humans and robots jointly handle objects; hand-guided robot teaching, where users physically demonstrate desired trajectories; assistive devices, such as powered prostheses and exoskeletons; and collaborative industrial robots designed to operate safely alongside human workers.

Physical Human–Robot Interaction presents distinct challenges compared to those encountered in traditional robotics. One of the most critical challenges is ensuring safety during physical contact. Because pHRI systems operate in direct contact with humans, unintended collisions, excessive forces, or unstable behaviors can result in injury. Addressing safety requires a combination of passive strategies, such as lightweight structures and compliant materials, and active strategies, including force sensing, real-time monitoring, and control algorithms that limit interaction forces and velocities.

Another important challenge is the realization of compliance and adaptability, as human users exhibit significant variability in strength, motion patterns, and responses to external forces. Rigid, position-controlled robots are poorly suited for such interaction, as they can resist human motion or amplify interaction forces. Consequently, pHRI systems often employ impedance or admittance control [50], series elastic actuation (SEA) [103], or

variable stiffness mechanisms to achieve compliant behavior while maintaining sufficient task performance.

Human intention estimation constitutes an additional challenge. Effective pHRI requires the robot to interpret human actions and intentions from incomplete and noisy sensory information, such as force measurements, kinematic data, or physiological signals. This inference must be performed in real time and under conditions of uncertainty, as human behavior can change rapidly in response to task demands or robot actions.

Finally, pHRI systems must address challenges related to usability and user experience, including transparency of robot behavior, predictability, and physical comfort. This is especially important in wearable and rehabilitation robotics, where the physical interaction may be continuous and prolonged, further increasing the importance of ergonomic design and long-term comfort.

1.2 Multi-robot systems

Multi-robot interaction is the research area that studies methods enabling multiple robots to influence one another's behavior in pursuit of individual or collective objectives. Among the numerous objectives that multi-robot interaction aims to there are reducing task completion times, providing redundancy that mitigates individual failures, and distributing sensing and actuation across wider spatial areas. In contrast to single-robot autonomy, where perception, decision-making, and control are primarily internal to one agent, multi-robot settings require explicit consideration of coupling across agents. This coupling can be intentional, as in cooperative task execution, or unintentional, as when robots share the same workspace, communication resources, or operational constraints. Multi-robot systems are utilized in various applications. Industrial environments frequently feature multiple manipulators operating concurrently within a workcell, sharing fixtures, workpieces, and temporal constraints imposed by process planning. Likewise, logistics and field robotics increasingly rely on fleets of mobile robots whose effectiveness depends on coordination in shared spaces and on collective management of tasks.

A key distinction in multi-robot interaction is between fixed multi-robot systems and mobile multi-robot systems. Fixed multi-robot systems consist of robots whose base positions are stationary, such as industrial manipulators. Typically, these systems operate in structured environments with a known geometry, stable architecture, favorable lighting conditions, and reliable communication.

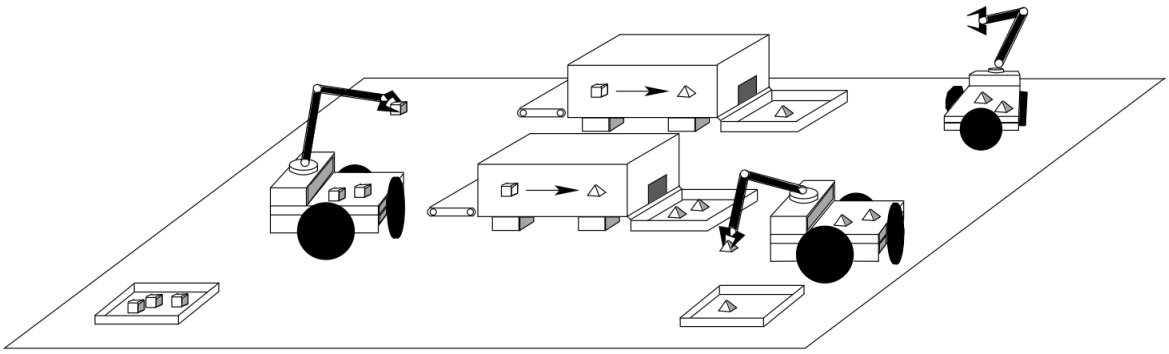


Figure 1.2 Multi mobile robot cooperating in industrial settings (Credits to: Wikimedia Commons).

Conversely, multi-mobile robot systems involve robots that move through an environment, such as wheeled ground robots, aerial vehicles, or underwater robots. These systems are often deployed in environments that are less structured and more dynamic, such as crowded areas. Mobile robots must handle navigation, localization, and obstacle avoidance while simultaneously coordinating with teammates to manage spatial interference, avoid congestion, and coordinate trajectories in real time. Warehouse logistics provides a common example of fleet-level coordination where a large number of robots must navigate shared corridors, manage congestion near storage locations and stations, and allocate tasks such as picking, replenishment, or delivery.

An important classification is team composition. Homogeneous teams consist of robots with similar sensing, actuation, and computational capabilities. Such teams support the development of scalable coordination policies and can simplify task allocation because members are largely interchangeable. On the contrary, heterogeneous teams comprise robots with diverse capabilities, such as combinations of aerial vehicles and ground vehicles, mobile robots and manipulators, or specialized sensors distributed across multiple platforms. Heterogeneity increases system capability but also increases the complexity of interaction because allocation and coordination must account for capability constraints, compatibility among robots, and potential dependencies between subtasks.

Multi-robot systems present several challenges that stem from their multi-agent nature. Inter-robot reliable and low-latency communication channels, required for effective management and control, are not guaranteed in all settings, especially in mobile system deployments. Motion coordination and traffic governance policies should be implemented to prevent deadlocks and potential robot-to-robot collisions. Fault handling and partial team losses can significantly impact the deployment's capabilities and must be addressed to minimize task

degradation. Cooperative perception and sharing of sensor data enhance the awareness of the robot team, but may be detrimental in cases where divergent world models exist.

All these aspects are emphasized by the number of system members. As the number of robots increases, coordination overhead can rise sharply, leading to an increased communication load that can saturate communication channels, reduce available space, and cause congestion due to unfeasible planning constraints, as well as an increased computational load.

1.3 Impedance Control

From the early stages of robotics, physical interaction has been one of the most crucial and widely studied topics in the field. Its relevance spans a broad range of applications, from industrial operations such as polishing, milling, and assembly, to human-oriented tasks like rehabilitation and assistance, where interaction with the human operator is paramount. As highlighted in [114], control strategies aimed at regulating the exchanged forces between a robot and its environment can be broadly categorized into two families: direct and indirect approaches.

Direct force control strategies act explicitly on the contact force, typically by closing a feedback loop at the force level. A widely known representative is hybrid force/motion control, introduced in [106], where the task space is decomposed into two orthogonal subspaces: motion-constrained directions and motion-unconstrained directions. The control variables are then selected accordingly, i.e., contact forces are regulated along constrained directions, while motion is controlled along unconstrained directions.

Conversely, indirect force control strategies do not require explicit loop closure at the force level. The most common approach in this category is impedance control, formalized for the first time in [50]. Rather than directly commanding either position or force, impedance control specifies the desired dynamic relationship between motion and contact force. In the Laplace domain, the impedance transfer function is defined as

$$Z(s) = \frac{F(s)}{V(s)}, \tag{1.1}$$

where s is the Laplace variable, and $F(s)$ and $V(s)$ denote otuput force and input velocity, respectively. In the operational space formulation, the most common impedance model is a

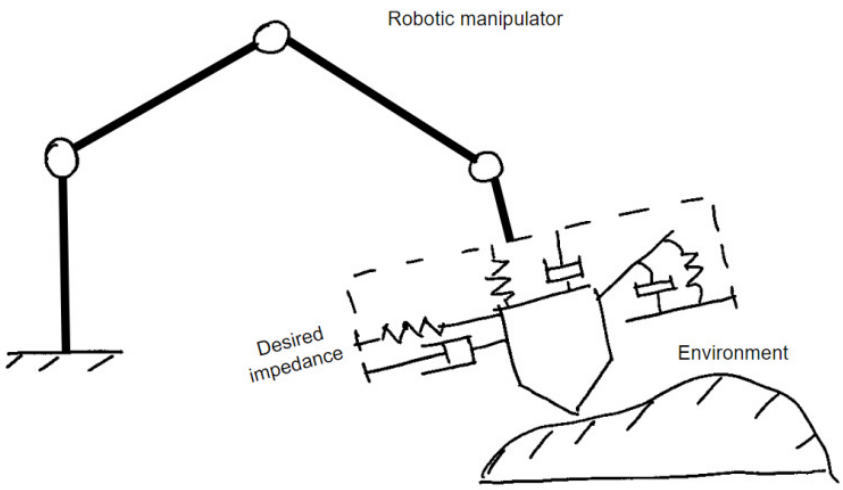


Figure 1.3 Cartesian Impedance Control in End-Effector Frame (Credits to: "Advanced Robotics for Manufacturing", by Emma Brandberg et al.)

second-order system,

$$F(t) = M\big(\ddot{x}(t) - \ddot{x}_v(t)\big) + D\big(\dot{x}(t) - \dot{x}_v(t)\big) + K\big(x(t) - x_v(t)\big), \tag{1.2}$$

which represents a mass–spring–damper behavior, with M, D, and K the desired inertia, damping, and stiffness, respectively, $x(t), \dot{x}(t), \ddot{x}(t)$ and $x_v(t), \dot{x}_v(t), \ddot{x}_v(t)$ the virtual/reference motion trajectory. In Laplace domain

$$F(s) = Z(s)(V(s) - V_v(s)) = (Ms + D + K/s)(V(s) - V_v(s)) \tag{1.3}$$

The stiffness term is often omitted when no target position is specified and free motion is desired.

Given the causality of (1.1), impedance control takes the robot motion state as input and produces a force (or wrench) command consistent with the chosen dynamic model. For this reason, it is primarily employed on torque-controlled robots. In typical implementations, the task-space output of the impedance law is mapped into joint torques through the Jacobian transpose, and combined with model-based terms to compensate for the robot dynamics.

The dual architecture, known as admittance control, can be obtained by inverting the causality of the impedance model. In admittance control, the external force applied to the robot is measured (or estimated) and fed to the controller, which generates a motion command. Admittance controllers are therefore particularly suitable for stiff position-controlled robots,

and more generally for platforms where direct torque control is not available, as is often the case for common industrial robots.

This duality is also reflected in the respective advantages and limitations. Impedance control is typically most effective in low-impedance environments and can render low inertia [50], whereas admittance control is better suited for high-impedance environments [5]. In [97], the authors propose a unified framework that embeds impedance and admittance behaviors as two extreme cases of a single controller family, enabling continuous interpolation and switching between them.

Classical impedance and admittance algorithms usually assume time-invariant parameters, yielding a fixed interaction behavior selected a priori. While effective in stationary conditions, this can lead to suboptimal performance when the environment changes or when the task comprises multiple phases, since the initial tuning may become too stiff (potentially unsafe) or too compliant (inaccurate).

Variable impedance control (VIC) generalizes impedance (and, by duality, admittance) control to address such limitations. In VIC, one or more adaptation laws are introduced alongside the impedance model to modify a subset of the dynamic parameters online, allowing the interaction behavior to change at runtime in response to task requirements or environmental variations.

The concept originated from human-robot cooperation, motivated by the observation that human impedance is modulated throughout a task; consequently, the robot impedance should adapt to the user's intention. In [52], one of the earliest variable impedance approaches is proposed for cooperative manipulation: a gain-scheduling strategy adapts damping based on robot velocity, and the impedance parameters are derived by characterizing the same cooperative task performed by two human operators. Since then, VIC has been extensively employed in physical human–robot interaction. In [40], a continuous adaptation law dependent on end-effector velocity is proposed to vary both inertia and damping, while exploiting kinematic redundancy to keep the apparent inertia close to the desired one. In [55], the variability of the admittance law is linked to direct and indirect human intention estimation, while [45] leverages electromyography signals to adapt admittance online.

Ensuring passivity under time-varying impedance/admittance parameters is a central theme in the literature, and a common solution relies on energy tanks [21]. First applied to variable admittance control in [38], and later extended in [39], energy tanks have also been incorporated via Port-Hamiltonian formulations to regulate passivity in the presence of parameter adaptation. In [118], an energy-tank-based variable admittance controller adapts

parameters when deviations from nominal behavior are detected, while [27] employs energy tanks to enforce passivity in hierarchical impedance control for redundant manipulators.

More recently, machine learning methods have increasingly been explored to learn adaptation policies for variable impedance/admittance control [3]. For instance, in [109], artificial neural networks are used to estimate a human–robot interaction model that is then exploited within a Model Predictive Control scheme to optimize impedance parameters online. In [67], Gaussian Processes are used to learn an impedance variation strategy, while in [113] the virtual inertia is directly produced by a feedforward neural network to adjust the admittance behavior online.

Few works adapt rheologic and tribologic models to realize admittance controllers. [110] exploits a combination of Maxwell and Voigt rheology models, [111] presents a controller based on the Zener rheology model while [44] proposes the use the LuGre friction model.

1.4 Mobile Manipulators

Mobile manipulators are robotic systems that integrate a locomotion platform with a manipulator arm (and typically an end-effector), enabling both mobility and dexterous physical interaction with the environment. By combining navigation capabilities with object-level manipulation, these systems aim to extend robotic autonomy beyond fixed workcells and structured factory floors to settings characterized by spatial extent, environmental variability, and heterogeneous tasks. As a result, mobile manipulators have become a central focus of research and development in service robotics, logistics, healthcare support, inspection and maintenance, and are increasingly being applied in flexible manufacturing and intralogistics.

Over the years, mobile manipulators have benefitted from significant advancements in both hardware and software in the field of robotics. These include developments in manipulator arms beyond standard industrial designs, such as redundant manipulators or mechanically compliant arms, as well as mobile bases with different kinematic configurations. These bases range from wheeled differential drive configurations to mecanum wheel-based systems, tracked bases, and even more advanced configurations like multi-copters [59] or submarine bases [95]. With the growing interest in legged robots, mobile manipulators using legged platforms are becoming increasingly common [74].

In general, the configuration space of a mobile manipulator (MM) is the composition of the configuration space of the manipulator arm, with dimension N_m, and the configuration space of the mobile base, which for planar bases typically used in industrial applications has dimension $N_b = 3$. The kinematic equations of an MM can be written as:

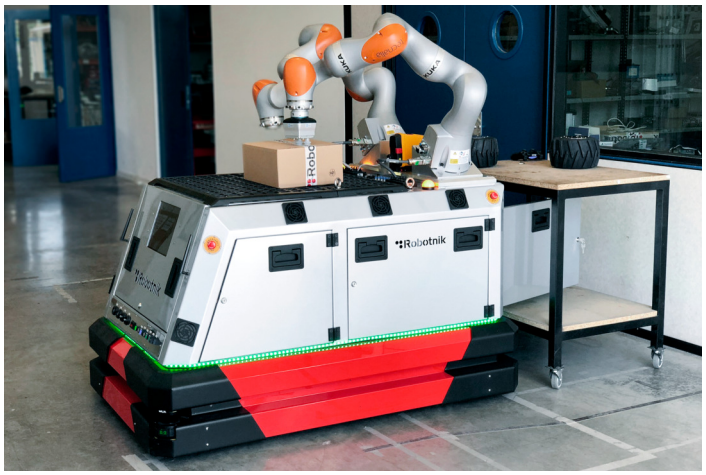


Figure 1.4 Industrial Mobile Manipulator (Credits to: Robotnik Automation S.L.)

$$\begin{aligned} \dot{x}_{ee} &= J(q)\dot{q} \\ \ddot{x}_{ee} &= \dot{J}(q)\dot{q} + J(q)\ddot{q} \end{aligned} \tag{1.4}$$

where $q = [q_b, q_m]^T$ represents the configuration of the system, with q_b denoting the configuration (pose $[x, y, \theta]$) of the mobile base and q_m the configuration of the manipulator. The vectors $\dot{q}$ and $\ddot{q}$ represent the time derivatives of the configuration. The Jacobian $J(q)$ is composed of the Jacobians of the mobile base $J_b(q_b)$ and the manipulator $J_m(q_m)$. Due to the common presence of non-holonomic constraints in mobile robots (differential kinematics, ackermann kinematics, etc.) the admissible instantaneous velocity command space of the mobile base may have dimensions smaller than N_b.

Mobile manipulator control strategies are typically divided into two main categories: decoupled control and whole-body control.

In decoupled control, the robot's components (mobile base and manipulator) are treated as separate entities, each with its own controller. Two separate controllers are used to compute the control actions for the base and the manipulator. This architecture is commonly used in industrial applications because of its simplicity, which also simplifies the risk assessments required by regulations. In pick-and-place tasks, for example, the base is moved close to the target object, and then the manipulator is moved to pick it up. The same process is repeated for placing the object. The position of the base can be chosen manually by operators or using reachability maps [129, 81]. Despite its widespread usage, decoupled control is still an active area of research. For example, in [20], explicit base control is used to perform manipulation while the base is in motion. In [100], a search-based grasping pipeline exploits kinematic

decoupling for robust visual servoing. In [107], the base position is optimized for a painting task to ensure constant velocity at the end-effector.

The alternative to decoupled control is whole-body control, where both the mobile base and manipulator are treated as a unified system, and control actions are computed for both simultaneously, typically through a single optimization problem. By considering the mobile manipulator as a single kinematic structure, redundancy resolution techniques can be applied to perform multiple tasks with strict or soft hierarchies [69] or weighted hierarchies [85, 23]. In these formulations, tasks are assigned priority levels, and lower-priority tasks are solved only within the null space of higher-priority tasks, ensuring that higher-priority tasks remain satisfied without interference.

In strictly hierarchical (lexicographic) task formulations, objectives are organized by priority, and feasibility and optimality at level k are enforced without degradation by tasks at higher levels. Lower-priority objectives are solved only over the null space induced by higher-priority tasks, exploiting redundancy only when higher-level constraints are satisfied. A classic approach that implements this structure is null-space projection, often used in conjunction with recursive redundancy resolution [117].

In more modern approaches, Hierarchical Quadratic Programming (HQP) has become a prominent method, where optimization problems for each layer of priority are solved sequentially, with each solution constrained by higher-level tasks [33]. In [49], HQP is used to implement differential inverse kinematics for mobile manipulators with a position-controlled arm and incomplete environment knowledge. Similarly, [93] combines the tracking performance of Prescribed Performance Control (PPC) with the redundancy-exploiting capabilities of HQP to ensure obstacle avoidance while respecting constraints. The work in [56] demonstrates the application of torque-based HQP to mobile manipulators with continuously changing tasks, considering nonholonomic constraints. More recently, a unified control framework for legged mobile manipulators that combines virtual model control with prioritized whole-body control has been proposed [126], which enables robust locomotion-manipulation coordination by explicitly accounting for arm-induced inertial effects.

Learning-based methods have also been explored for whole-body control. For example, [51] presents a Reinforcement Learning (RL) framework for training whole-body control policies in mobile manipulators, leveraging causal dependencies in the reward function. In [43], RL with Advantage Mixing is employed to exploit causal dependencies in the action space and regularized online adaptation to achieve effective Sim2Real transfer. A neural network architecture for optimizing coordinated navigation and manipulation in mobile manipulators is proposed in [127]. Additionally, [64] introduces a whole-body impedance

controller with a radial basis neural network for estimating dynamic uncertainties during task execution.

Safety and compliance have become critical aspects of mobile manipulator control, especially in human-robot collaboration settings. For example, techniques like variable impedance control have been used to modulate the robot's behavior in response to dynamic changes in the environment or task execution [3, 55]. The use of energy tanks in hierarchical control structures has also gained attention as a means of ensuring passivity in safety-critical applications [27, 21].

In the context of nonholonomic mobile manipulators, these methods have been adapted to maintain both safety and performance, particularly when the base's motion is constrained by nonholonomic dynamics. These formulations allow for the incorporation of compliant behaviors even in the presence of significant constraints, facilitating safe and efficient interaction in unstructured environments [113, 67].

1.5 Cooperative manipulation

A frequently cited application of physical human–robot interaction (pHRI) is collaborative transportation, i.e., the cooperative manipulation and transport of bulky payloads that are difficult to move safely and efficiently by either a single human or a robot alone. Research contributions on this topic have been numerous over the years.

Safe physical interaction typically requires compliant behavior under interaction wrenches and, in many cases, interaction force/wrench measurement or estimation. For this reason, impedance and admittance controllers are commonly used as the core of cooperative manipulation frameworks.

A seminal work on cooperative manipulation is [52], which introduced a variable impedance controller for object transportation, with controller tuning informed by impedance observed during human–human cooperative transport. In [60], the authors empirically demonstrated performance advantages of impedance control with respect to alternative control approaches. An impedance controller with varying damping based on the time derivative of the interaction force was presented in [29]. The integration of visual and haptic information has been investigated in [6, 128]: in the former, visual sensing is used to estimate the task state, whereas in the latter it is used to estimate the human control input. A motion-capture systems is used, instead of vision, in [115] to implement a hierarchical quadratic programming (HQP)-based architecture for deformation-agnostic collaborative transportation. Game-theoretic approaches have been adopted to address role arbitration in [41] and to

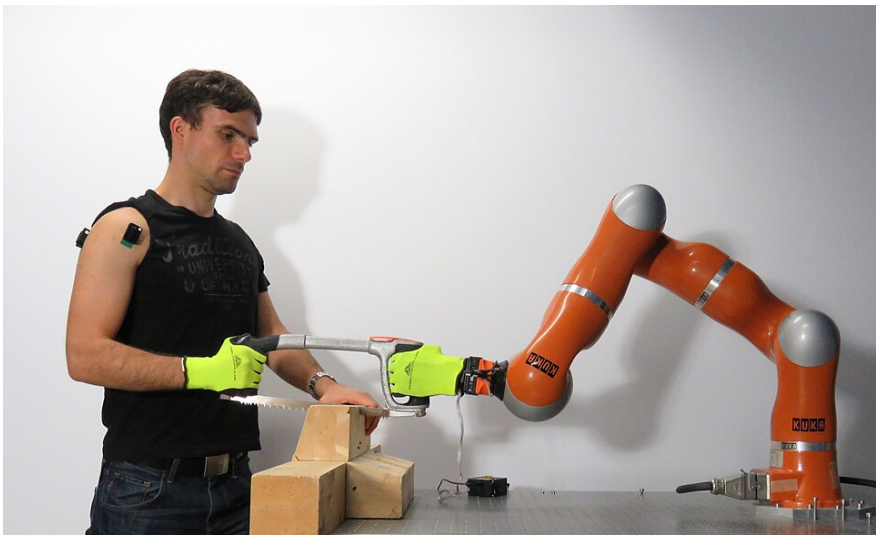


Figure 1.5 Advanced cooperative manipulation of a saw (Credits to: Peternel L. [101]).

design distributed model predictive controllers in [42]. The transport of deformable objects is tackled in [92] by estimating the deformation of the payload using depth images and a Convolutional Neural Network to produce a suitable twist reference to compensate changes in deformation.

A relevant subproblem within collaborative transportation is cooperative manipulation with mobile manipulators, which enables object transport across large workspaces. Representative mobile-manipulator frameworks include [94], which applies decoupled impedance controllers to a two-wheeled mobile manipulator; [14], which integrates energy tanks and control barrier functions within a whole-body control approach to improve safety; [119], which proposes a multi-modal architecture based on multiple HQP problems; and [99], which estimates object dynamic parameters online to adapt a Cartesian impedance controller; [80], which embed a disturbance-aware discrete Riccati Algebraic Equation into an MPC architecture to handle human uncertainty.

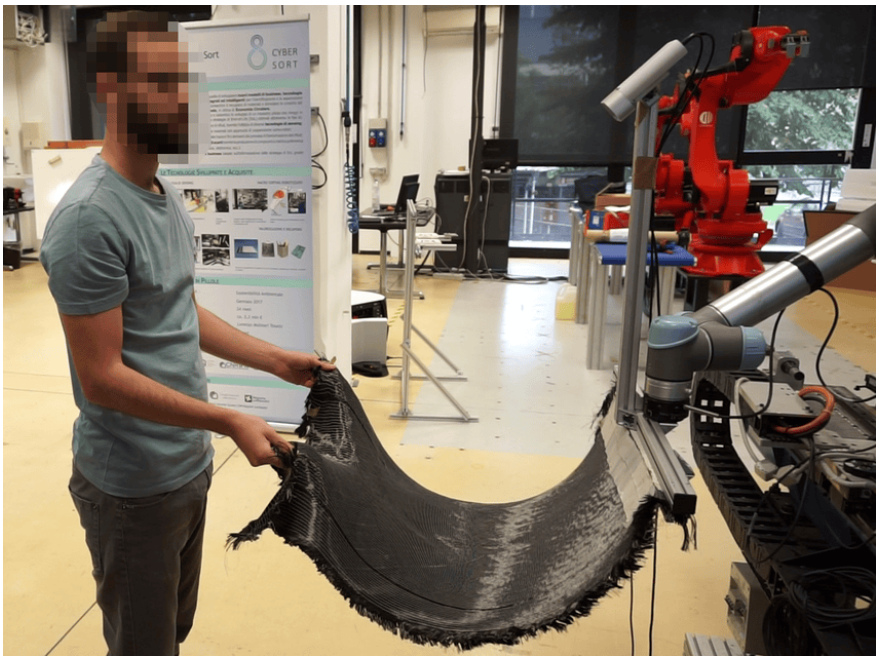


Figure 1.6 Collaborative Transportation of a soft material.

1.6 Formation planning and control

In a multi-robot system, a common objective is to maintain a prescribed geometric configuration while moving through the environment [7]. Such formations arise in cooperative object transport [131, 122], entertainment and drone shows [90, 102], patrolling and perimeter surveillance [58, 54], and vehicle platooning [70, 68]. While formation control addresses how the agents regulate relative geometry, operating in cluttered or dynamic environments typically also requires planning to generate collision-free motions that respect formation and vehicle constraints.

Formation control strategies are commonly grouped into three families [11]: virtual-structure, behavior-based, and leader–follower approaches. In virtual-structure methods [123, 66, 24], the team is treated as a single entity whose kinematics and dynamics are defined at the formation level. The resulting "structure motion" is then mapped into per-agent references, which are tracked by local controllers. This perspective provides coherent, globally consistent motion generation, but it can be less flexible when the formation must adapt locally (e.g., in tight passages) unless additional mechanisms are introduced.

Conversely, behavior-based approaches [10, 46, 65] design a set of behaviors for each agent (e.g., goal seeking, cohesion, separation, obstacle avoidance), each producing a partial command or reference. These are combined to obtain the final control input, and the team

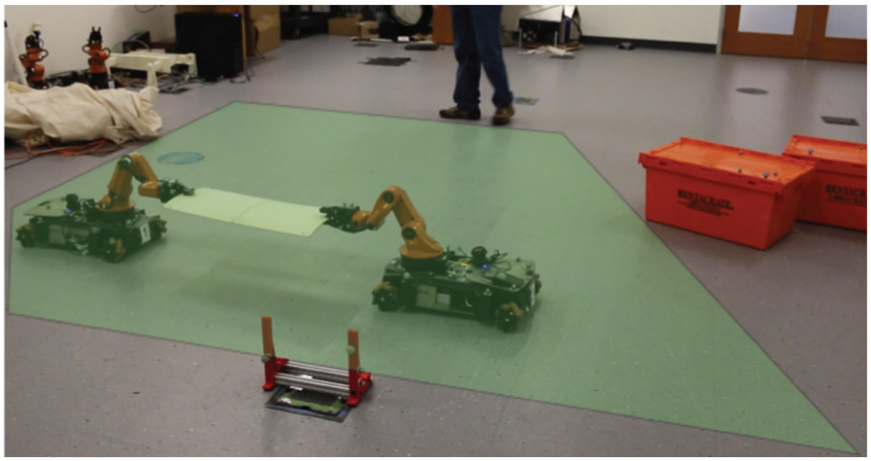


Figure 1.7 Formation control for transportation of components (Credits to: Alonso-Mora J., et al. [8])

behavior emerges from local interactions. Such methods are typically modular and reactive, which is advantageous in uncertain environments, but they may require careful tuning to avoid undesirable emergent effects (e.g., oscillations or deadlocks) in dense scenarios.

In leader–follower approaches [112, 48, 16], each agent is assigned a role as leader or follower. Followers track the leader's motion while maintaining a desired relative offset, while the leader tracks a reference trajectory. This architecture is conceptually simple and often effective in practice, but performance and robustness can depend strongly on leader selection and communication/sensing assumptions, especially under occlusions or intermittent connectivity.

Blended strategies combine these families to exploit complementary strengths. For example, [13] adopts a virtual-structure formulation while equipping each agent with local behaviors to enable obstacle avoidance. In [132], a hybrid leader–follower/virtual-structure strategy is developed for AUV formations, aiming to maintain cohesion with limited communication while improving obstacle avoidance. Similarly, [125] proposes a hierarchical fuzzy controller for mecanum-wheeled robots that combines leader–follower structure with behavior-based collision avoidance.

From the perspective of path planning for formations, an orthogonal and practically relevant classification is based on the decision-making architecture [4], which groups methods into centralized, decentralized, distributed, and hybrid schemes. This distinction is important because it directly affects scalability, communication requirements, and the types of guarantees (e.g., completeness or safety margins) that can be provided.

In centralized approaches, a single planning authority (an on-board coordinator or an external unit) computes plans for the entire team. Representative examples include potential-field methods [25], where planning may be performed for a single virtual target (e.g., the formation centroid) and then converted into individual references, offering reactivity but suffering from classical issues such as local minima. Sampling-based planners have also been adapted to formation constraints: [57] applies probabilistic roadmaps (PRMs) to permutation-invariant formations, while [8] associates roadmap nodes with convex regions of free space and uses constrained optimization to validate feasible formation transitions between regions. Similarly, [18] proposes an RRT*-based planner that samples a reduced-dimensional formation space (centroid, orientation, scaling) and then maps the resulting path to per-robot trajectories, providing flexibility (e.g., squeezing through narrow gaps) while shifting complexity to transformation and collision-checking. Finally, optimization-based planners [62] can directly encode collision avoidance and smoothness constraints (e.g., via mixed-integer formulations), offering strong constraint handling but potentially incurring higher computational cost.

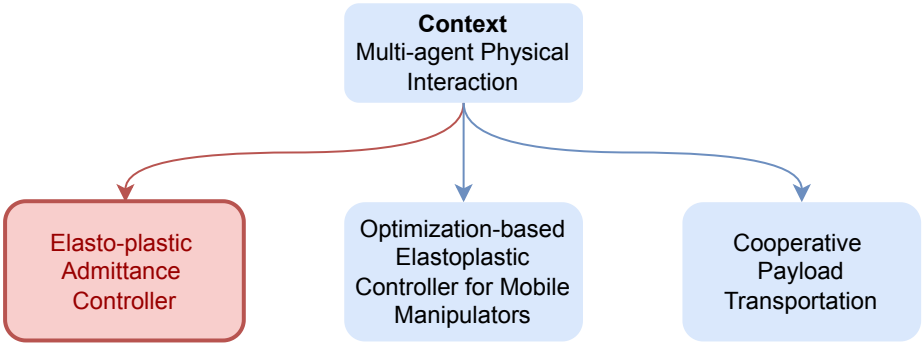
In decentralized approaches, each robot computes its motion locally using on-board information and limited communication, typically restricted to nearby neighbors. For instance, [83, 84] propose potential-field planning with neighbor-to-neighbor information exchange to limit communication overhead, enabling reactive coordination but remaining sensitive to local minima and oscillatory behaviors in crowded settings. Grid-based methods such as cellular automata [53] provide simple local update rules that can yield coordinated motion, though discretization can limit path quality and complicate kinodynamic constraints. Decentralized sampling-based methods have also been studied; [26] presents a decentralized multi-agent RRT approach that emphasizes scalability under partial information, at the expense of requiring mechanisms to manage inter-agent conflicts and coordination.

In distributed approaches, robots cooperatively solve a global planning or coordination task without relying on a single central unit, typically through structured information sharing or negotiation. Priority-based coordination is a classical example: [12] improves decoupled planning by searching for priority schemes that increase solvability and reduce path length. Visibility-graph-based cooperation has also been explored; [73] uses geometric constructs to generate intermediate objectives that guide cooperative planning, which can be effective when environment geometry is reliably known but may be less straightforward under uncertainty or partial maps. Heuristic and bio-inspired strategies have been proposed as well; [34] combines evolutionary programming with artificial bee colony optimization in unknown environments, trading theoretical guarantees for flexibility and empirical performance.

Hybrid architectures combine elements of the above to balance global coordination and local responsiveness. For example, [130] proposes a cloud-based, decoupled navigation system in which a "system manager" constructs a global cost map using multi-criteria decision making and then performs global planning and coordination, while local agents execute and react to changes. In [98], robots collaborate by sharing intermediate solutions and dynamically forming teams when path conflicts arise, explicitly addressing limited communication while aiming to maintain progress in complex environments.

Chapter 2

Elastoplastic Controller for Interaction



This chapter presents an interaction control approach that allows a position-controlled manipulator to demonstrate both compliant guidance and precise task execution. The proposed method introduces an elasto-plastic behavior that generates motion deformation, which is added to the nominal task reference. When interaction forces are small, the robot responds elastically, resulting in reversible deviations that disappear once the external interaction ceases. However, when the interaction exceeds a breakaway threshold, the robot transitions smoothly to a plastic regime, enabling the user to make lasting modifications to the motion without generating restoring forces.

The model is based on friction-inspired internal dynamics and is tailored for effective physical human-robot interaction. The dynamics is extended to ensure a clearly defined elastic region, prevent unintended recovery during plastic motion, and maintain isotropic behavior in three-dimensional space. Additionally, a dedicated mechanism manages direction reversals during plastic motion, preventing oscillations and abrupt switching. A reset condition, based on ongoing interaction activity, restores elastic behavior once the interaction stops, allowing for future guidance actions while maintaining the desired permanent deformation.

The implementation details and experimental validation conducted on a collaborative manipulator show that the method produces predictable and intuitive behavior for users. It supports both transient and persistent corrections, remaining compatible with standard robot control systems based on position references.

The Chapter is structured as follows:

- **Section 2.1** provides the definition of the guidance-accuracy trade-off in physical interaction and design goals for an elasto-plastic interaction model.
- **Section 2.2** reviews the fundamentals on which the method presented is based on
- **Section 2.3** introduces internal dynamics used to reproduce elastic behavior under small interactions and plastic behavior beyond a critical strain or breakaway region. Then it presents the extension to support multiple dimensions.
- **Section 2.4** discusses the reversal-related issues and presents a mechanism that preserves predictable plastic evolution during direction changes. Then it provides a reset strategy to restore elastic behavior when interaction becomes negligible, supporting repeatable guidance cycles
- **Section 2.5** describes the integration in a robotics software stack and presents an experimental evaluation on an industrial robot with force/torque sensing.
- **Section 2.6** concludes the chapter and summarizes the methodologies presented and the results.

The content of this Chapter is based on the work in [37].

2.1 Introduction

2.1.1 Context

Physical human-robot interaction (pHRI) and, more generally, contact-rich tasks represent a cornerstone of modern robotic applications. These tasks span a wide range of domains, including industrial operations such as assembly, milling, and material handling, as well as medical procedures such as surgery, rehabilitation, and patient-assistive robotics. In all these scenarios, the robot must not only execute precise motions but also continuously adapt its behavior to the surrounding environment and to the forces exerted by interacting agents.

Ensuring safe and effective interactions is critical: excessive forces can damage sensitive robotic components, compromise task performance, or, more importantly, pose a safety risk to human operators. Therefore, the ability to modulate compliance in real time is a fundamental requirement in these applications.

The necessary compliance can be achieved either through dedicated hardware, such as Series Elastic Actuators (SEAs) [103] or Remote Center of Compliance (RCC) devices [114], or by means of active control strategies. Among these, impedance and admittance control are the most widely adopted. Impedance control regulates the forces generated by the robot in response to deviations from a desired trajectory, effectively shaping the mechanical impedance of the system. Admittance control, on the other hand, is particularly suited to stiff, position-controlled manipulators often found in industrial settings. It generates a reference trajectory for the robot's end-effector based on measured external forces, typically using a second-order dynamic model equivalent to a mass–spring–damper system. This approach allows the robot to respond to interaction forces in a controlled and predictable manner, effectively rendering a desired compliance without the need for inherently compliant hardware.

Different tasks naturally benefit from different dynamic behaviors. Tasks that require precise trajectory tracking, such as machining or assembly, benefit from a dominant elastic term, which ensures that the end-effector closely follows the desired path. Conversely, tasks that involve manual guidance or human-led positioning often require minimal or no elastic contribution, allowing the robot to move freely under the operator's influence while maintaining a degree of stability. In practice, these different behaviors can coexist within a single application: a collaborative robot may need to switch seamlessly between precise, high-stiffness operations and compliant, human-guided motions. Achieving this versatility requires control strategies that can dynamically modulate the robot's response characteristics in real time.

2.1.2 Contribution

Drawing inspiration from rheology, the branch of mechanics that studies material deformations under force, several nonlinear dynamic models can be employed to reproduce a wide range of mechanical behaviors, including both elastic and plastic responses. Building on this concept, this chapter introduces a dynamic model capable of representing two primary behaviors: an elastic behavior, which produces temporary, reversible changes in the end-effector position, and a plastic behavior, which introduces permanent offsets in the trajectory

when interaction forces exceed a certain threshold. The transition between these behaviors is governed by the magnitude of the applied force, enabling the robot to adjust its compliance automatically in response to task demands.

To implement this behavior, the proposed dynamics utilize the LuGre friction model, a one-dimensional model well-known for capturing both pre-sliding (elastic) and sliding (plastic) effects. By incorporating these characteristics, the model naturally embeds the desired elasto-plastic behavior into the robot's response. Building on this foundation, a variable-admittance Cartesian control strategy is proposed for position-controlled manipulators, referred to as the elasto-plastic controller. In this framework, the offset generated by the dynamic model is extracted from the measured contact forces at the robot's end-effector and incorporated into the reference trajectory, effectively allowing the robot to transition smoothly between elastic and plastic behaviors while maintaining stability and precision.

The proposed controller has been implemented as a ROS 1 controller and validated experimentally on a UR10e collaborative robot. The results demonstrate the system's ability to provide compliant yet precise responses, facilitating both human-guided movements and tasks that require high positional accuracy. This approach represents a significant step toward versatile, safe, and adaptive physical human-robot interaction, capable of handling a broad spectrum of contact-rich applications.

2.1.3 Assumptions

Throughout the work, the following assumptions have been made:

- The robot can be controlled in position, with low-level controllers having sufficiently high bandwidth to perform accurate reference tracking
- A force-torque sensor is mounted at the wrist to measure the contact forces, and can provide reliable force measurements
- The robot can move freely along the three linear axes of the Cartesian space. Although only linear movements are considered, the admittance model should also be able to handle rotations. The mixture of linear and rotational movements is not addressed

These assumptions are easily verified in an industrial setting as they describe a large variety of industrial setups.

2.2 Fundamentals of Physical Human-Robot Interaction Model

Physical human-robot interaction (pHRI) studies the collaboration between humans and robots where forces are exchanged to perform shared tasks, such as co-transport, assembly, or rehabilitation. Standard models in pHRI include impedance and admittance control frameworks, which define the robot's dynamic response to human-applied forces, enabling compliant and predictable behavior. Shared control and cooperative control models coordinate the contributions of both humans and robots, striking a balance between autonomy and human guidance to accomplish tasks efficiently and safely. Additionally, predictive and adaptive models leverage human motion and force estimation to anticipate intent, enabling smooth, intuitive, and safe physical collaboration. These approaches are crucial for designing robots that can interact safely and effectively in close physical proximity to humans.

The Section will shortly report two models on top of which the Thesis contributions have been developed.

2.2.1 Admittance Control Formulation

In physical human-robot interaction (pHRI), humans and robots exchange forces to perform tasks such as co-transporting objects or collaborating on assembly. An admittance controller maps external forces applied to the robot into desired motion, allowing the robot to respond safely and predictably to human input. The basic admittance control can be modeled as a mass-damper(-spring) system and considers a position-controlled robot subject to a Cartesian trajectory.

Mathematical formulation

The controller takes as input the force applied to the end-effector, F_h, and the robot's actual Cartesian kinematic state, and computes an offset, x_d, from the actual trajectory. To do that, the controller implements the simple Cartesian dynamic model:

$$M\ddot{x}_d(t) = F_h(t) - F_r(t) \tag{2.1}$$

where $M = diag(m_x, m_y, m_z)$ is the chosen inertia matrix, $F_h(t)$ is the force applied on the robot by external agents (e.g. the human operator) and $F_r(t) = f(\dot{x}(t))$ is a resistant force computed by a dedicated admittance model which exhibits the required elasto-plastic

properties. The offset is easily obtained by reordering (2.1) and integrating twice with respect to time.

The actual Cartesian reference x_e provided to the robot is

$$x_e(t) = x_r(t) + x_d(t) \tag{2.2}$$

where x_r is a reference trajectory.

If the robot remains in a stable position, i.e., no trajectory is being executed, $x_r = 0$, and the external agent fully controls the position. In the Laplace domain:

$$X_e(s) = X_d(s) = \frac{F_h(s) - F_r(s)}{s^2 M} \tag{2.3}$$

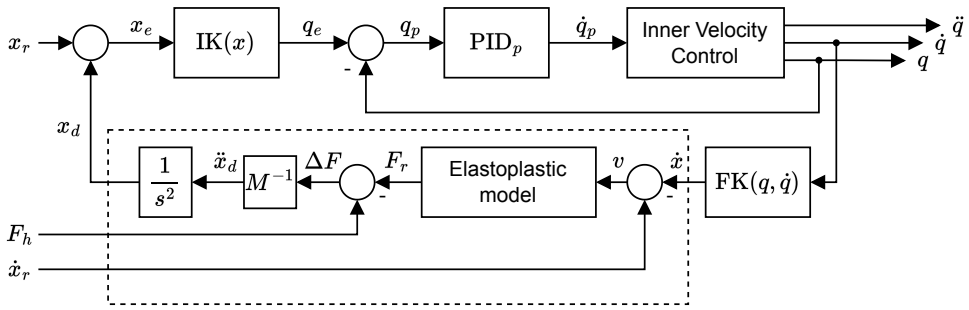


Figure 2.1 Control scheme considered. The elasto-plastic controller is implemented in Cartesian space and computes a deformation x_d from the actual trajectory.

Limits and Discussion

An admittance controller, however, reveals several practical limitations to this approach, especially when you try to guide a robot manually. The natural approach is to reduce the stiffness K as much as possible. In this case, the robot becomes very compliant, making it easy to move along the desired trajectory. This is great for safety and comfort, because it does not require push against a stiff robot. However, this introduces some important limitations. First, as the stiffness goes to zero, the robot loses its ability to follow a precise path. Any small disturbance, actuator friction, or slight human-induced deviation can cause the robot to deviate from its intended trajectory. This is a problem, for example, in assembly tasks where precise positioning is required.

Second, even if the stiffness is low, the robot's mass and damping still resist motion. If damping is too high, the robot feels sluggish; if too low, it can oscillate while you guide it. Achieving smooth, intuitive motion while maintaining stability is not a trivial task.

Ultimately, there is an inherent trade-off between admittance control and task precision in terms of ease of guidance. Easiness of motion requires low stiffness, thus sacrificing precision. Conversely, tracking precision requires high stiffness, resulting in increased guiding difficulty. For this reason, in practice, many applications use variable or adaptive admittance control, where stiffness is low during guiding phases and increased during precision tasks, or combine impedance with admittance strategies to maintain both compliance and accuracy.

Limitation	**Explanation**
Low Stiffness $\rightarrow$ High Compliance	Robot is easy to guide, but loses the ability to precisely follow a trajectory.
Task Accuracy	With near-zero stiffness, the robot cannot exploit its inherent precision for tasks like assembly.
Sensitivity to Disturbances	Low stiffness makes the robot more prone to oscillations and deviations caused by human-induced motion or external perturbations.
Guidance vs. Precision Trade-off	Cannot simultaneously achieve maximum ease of guiding and maximum precision; some compromise is always required.
Damping Effects	Even with low stiffness, mass and damping affect motion; too much damping feels sluggish, too little can cause instability.

Table 2.1 Limitations of Impedance Control in Human-Guided Tasks

2.2.2 Elasto-Plastic Formulation

As discussed above, in human-guided robotic tasks, such as manually positioning a robot for assembly operations, conventional impedance control presents an inherent trade-off. A low stiffness setting facilitates easy guidance but compromises positional accuracy, whereas high stiffness enhances precision but impedes intuitive human motion. Elasto-plastic control provides an effective approach to address this challenge.

This control strategy is inspired by the mechanical behavior of elasto-plastic materials. In the elastic region, the robot behaves like a spring, automatically correcting small deviations from the desired trajectory to preserve accuracy and stability. In the plastic region, when the

applied force exceeds a predefined threshold, the robot yields, offering minimal resistance and allowing the human operator to guide it with ease.

The advantages of this approach are significant. First, a facilitated guidance: When the human applies a force exceeding the threshold, the robot transitions into the plastic phase, reducing resistance and enabling smooth, intuitive motion. Second, a retention of precision: Minor deviations remain within the elastic region, enabling the robot to correct small errors automatically, which is critical for high-precision tasks such as object assembly. Third, intuitive and safe interaction: The combination of elastic and plastic behavior provides continuous feedback to the human operator, allowing freedom of motion when needed, and promotes both safety and usability. Finally, the adaptability: By adjusting the threshold separating the elastic and plastic phases, the robot's compliance can be tuned according to task requirements, achieving a desirable balance between guidance and precision.

Despite its advantages, elasto-plastic control requires careful parameter tuning. The stiffness, damping, and yield threshold must be selected to ensure smooth transitions between phases and to avoid oscillatory or jerky behavior. Overall, elasto-plastic control provides a systematic framework for balancing compliance and precision in human-guided robot tasks, effectively mitigating the limitations of traditional impedance control.

In such a field, an extension of the common rheological LuGre model is investigated to address the intuitive physical human robot interaction.

LuGre Rheological Model

A well-known model that presents a mixture of elastic and plastic behaviors is the LuGre friction model [22]. Developed for control purposes, it represents the contact between two bodies as bristles, with the friction force depending on the average deflection, as shown in Figure 2.2. The model can be formulated as:

$$\begin{cases} F_r &= \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v \\ \dot{z} &= v - |v| \frac{\sigma_0}{g(v)} z \\ g(v) &= f_c + (f_s - f_c) e^{-(v/v_s)^2} \end{cases} \tag{2.4}$$

where $z \in \mathbb{R}$ represent the average bristle deformation, $v \in \mathbb{R}$ is relative velocity between the bodies in contact, σ_0, σ_1 are the stiffness and damping coefficients of the bristle, σ_2 is the viscous friction coefficient, f_c is the coulomb friction force, f_s is the static friction force and v_s is the Stribeck velocity.

The first equation represents the friction force generated during the motion. It is the sum of

the bristle deformation effect $\sigma_0 z + \sigma_1 \dot{z}$ and the viscous friction $\sigma_2 v$. The second equation describes the bending of the bristles. It can be seen that

$$\lim_{z \to \bar{z}_{ss}} \dot{z} = 0 \tag{2.5}$$

where $\bar{z}_{ss} = \frac{g(v)}{\sigma_0} sign(v)$. Therefore, the bristle has maximum deflection and maximum elastic force.
Finally, $g(v)$ represents the velocity-dependent steady-state friction level, described by the Stribeck curve [72]. For the scope of this work, the Stribeck term is dropped to simplify the model and remove unwanted behaviors.

Limits and Discussion

Elasto-plastic control offers a way to combine compliance and precision in human-guided tasks, but requires attention during the implementation. One of the main limitations lies in the threshold tuning: the transition between the elastic and plastic phases must be carefully set. If the threshold is too low, the robot yields too easily and loses precision; if it is too high, the robot becomes stiff and harder to guide. Related to this is the smoothness of transitions between the two phases. Abrupt or poorly tuned transitions can make the robot feel jerky or unpredictable, which can negatively affect the safety and comfort of the human operator.

Another important consideration is the complexity of implementation. Unlike standard impedance control, which relies on a single stiffness-damping relation, elasto-plastic control requires algorithms to detect when to switch phases and to manage the robot's behavior in each region. This makes the system more sensitive to the operator's applied forces: rapid or highly variable inputs can cause the robot to switch unexpectedly between elastic and plastic behavior, reducing predictability. Finally, the robot's performance depends strongly on the careful tuning of stiffness, damping, and the yield threshold; improper tuning can compromise both safety and task accuracy.

All these points will be addressed in the following description of the proposed control strategy.

Table 2.2 Comparison of Impedance and Elasto-Plastic Controllers in Human-Guided Tasks

Aspect / Limitation	**Impedance Controller**	**Elasto-Plastic Controller**
Ease of Guidance	Easy if stiffness is low, but loses accuracy	Easy to guide when force exceeds threshold (plastic phase)
Precision	High if stiffness is high, low if stiffness is low	Maintains precision in the elastic phase, allows yielding for guidance
Parameter Tuning	Single set of stiffness/damping parameters; simpler to tune	Basic impedance parameters plus additional ones that regulates mode transitions.
Transition Smoothness	No transition	Proper tuning provides smooth transitions
Predictability	Predictable linear response	Predictable elastic and plastic responses.
Implementation Complexity	Relatively simple	Additional mathematical complexity required to provide different responses
Sensitivity to Disturbances	Low stiffness $\rightarrow$ sensitive, high stiffness $\rightarrow$ stiff	Elastic region resists small deviations, plastic region allows yielding; transitions sensitive

2.3 Elasto-Plastic Model

According to the consideration in Section 2.2.2, a LuGre-based control strategy for pHRI has been investigated. The high-level schema is reported in Figure 2.2.

2.3.1 LuGre's extension to model Elastic Displacement

Among the different characteristics exhibited by friction models, two properties are particularly important for developing the elasto-plastic model: pre-sliding displacement and stiction.

Pre-sliding displacement refers to the small, reversible motion that occurs when the applied force remains below a breakaway threshold. In this regime, the contact behaves as an elastic element: the system undergoes a temporary deformation that disappears once the force is removed. Stiction, or static friction, describes the complementary situation in which forces below the breakaway threshold do not produce any permanent motion. Although the system may still deform elastically, no irreversible displacement (i.e., slip) occurs [30].

For the elasto-plastic model, both properties are essential. During the elastic phase, the system should deform without introducing permanent offsets in the reference trajectory, which requires the coexistence of reversible deformation (pre-sliding) and the absence of creep or drift (stiction). However, the standard LuGre model intrinsically reproduces only

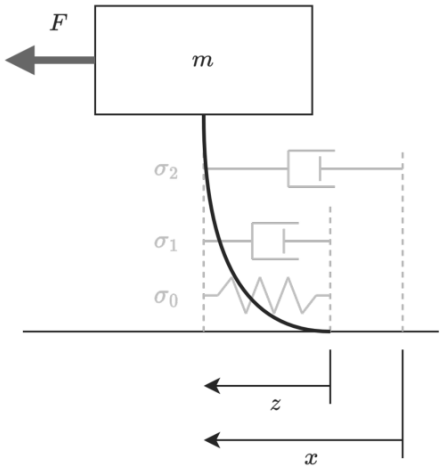


Figure 2.2 Representation of the LuGre friction model.

the pre-sliding displacement: while it captures the elastic-like deformation preceding slip, it does not guarantee true stiction. Under sustained forces below the breakaway threshold, its internal state may drift slowly, resulting in small but irreversible motions.

To ensure that the model exhibits genuine elastic behavior below the breakaway threshold, the LuGre formulation must therefore be modified to explicitly enforce stiction. A widely adopted modification, introduced in [30], alters the internal dynamics of the LuGre state to prevent drift when the applied force is insufficient to trigger slip. The resulting model preserves pre-sliding deformation while guaranteeing no permanent displacement, thus yielding the desired elastic response prior to the onset of the plastic phase.

$$\begin{cases} F_r &= \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v \\ \dot{z} &= v - \tilde{\alpha}(z,v)|v|\frac{\sigma_0}{f_c} z \end{cases} \tag{2.6}$$

with the function α defined as $\alpha : \mathbb{R} \to [0,1]; z \mapsto \alpha(z)$

$$\alpha(z) = \begin{cases} 0 & |z| < z_{ba} \\ \frac{1}{2}\sin\left(\pi \frac{z - \left(\frac{z_{ss}+z_{ba}}{2}\right)}{z_{ss}-z_{ba}}\right) + \frac{1}{2} & z_{ba} \le |z| < z_{ss} \\ 1 & |z| \ge z_{ss} \end{cases} \tag{2.7}$$

$$\tilde{\alpha}(z,v) = \begin{cases} \alpha(z) & v \cdot z > 0 \\ 0 & v \cdot z \le 0 \end{cases} \tag{2.8}$$

and $z_{ss} = \frac{f_c}{\sigma_0}$. A plot of $\alpha(z)$, with $z_{ba} = 0.5$ and $z_{ss} = 1.5$ is shown in Figure 2.3
In the model, for $|z| < z_{ba}$, $\tilde{\alpha}(z,v) = 0$ implies $\dot{z} = v$ and results in the whole motion translated into bristle deformation. For growing z, with $|z| > z_{ba}$, the system behavior approaches the original LuGre model. In Equation (2.8), the second case allows the return into the stiction phase in case of motion reversal.
Figure 2.4b shows the different behaviors with and without $\alpha(z)$. For an admittance controller, the return into pure stiction phase due to motion reversal is undesirable as it can limit the usability of the controller. For this reason, the second case of (2.8) is neglected and $\bar{\alpha}(z,v) = \alpha(z)$ will be used. More details on motion reversal and plastic-to-elastic evolution are provided in Section 2.4.

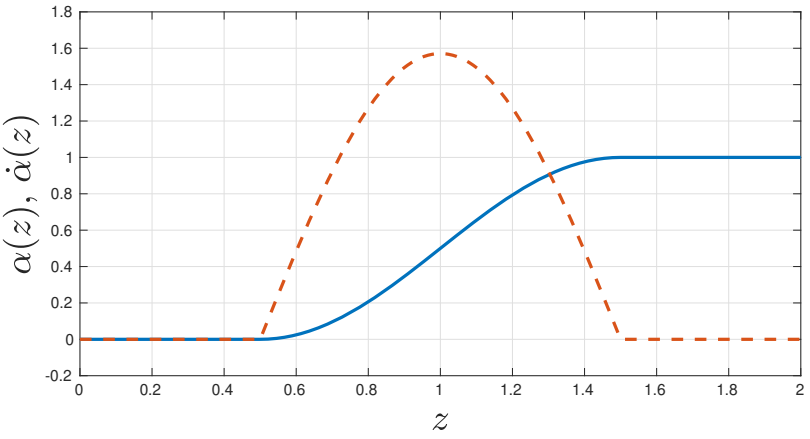
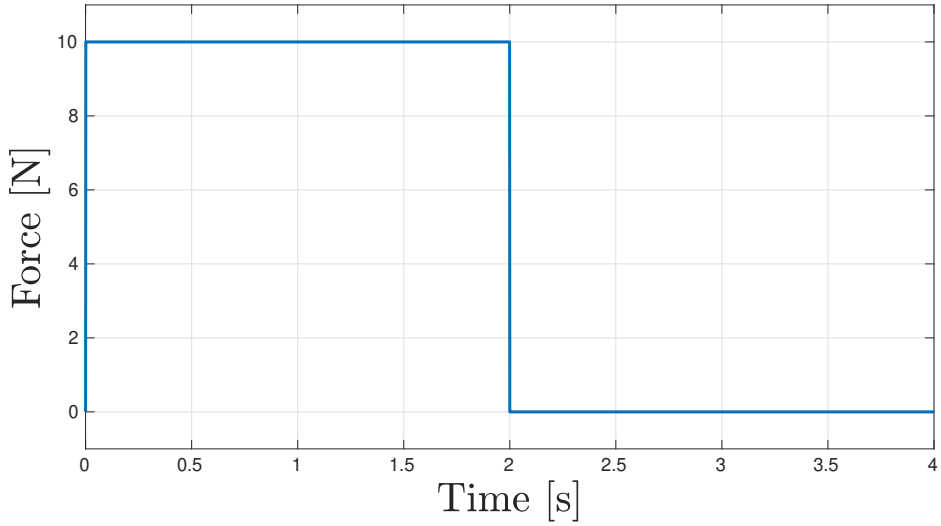
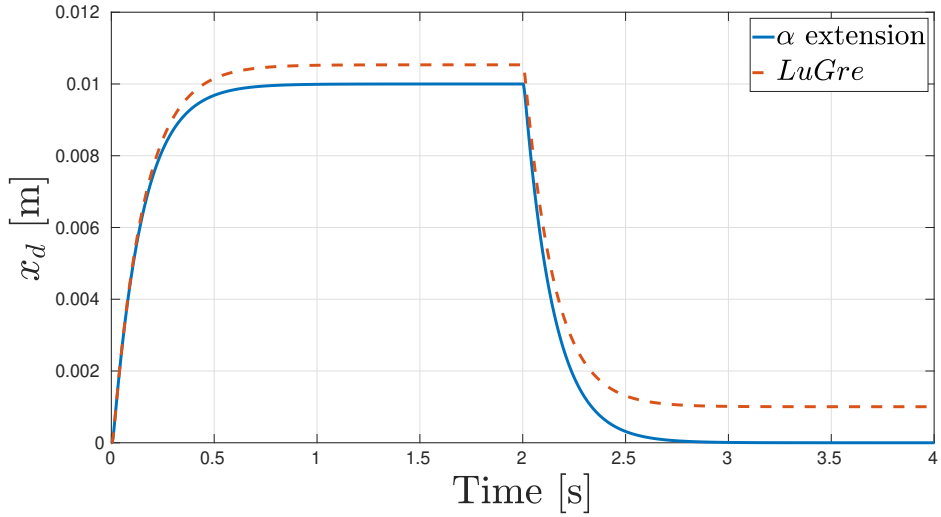


Figure 2.3 Plot of functions $\alpha(z)$ (blue line) and $\dot{\alpha}(z)$ (dashed red line). $z_{ba} = 0.5$, $z_{ss} = 1.5$.



(a) Input force



(b) Elastic displacement computed with (blue line) and without (red dashed line) $\alpha(z)$.

Figure 2.4 Elastic response of LuGre model extended with stiction

2.3.2 Plastic Displacement Analysis

Consider system (2.6). For $\alpha(z) > 0$, the force produced by the model is:

$$F_r = F_{\text{elastic}} + F_{\text{viscous}} = \sigma_0 z + (\sigma_1 \dot{z} + \sigma_2 v), \tag{2.9}$$

which corresponds to the classical LuGre formulation: an elastic contribution proportional to the internal state z, and a viscous contribution associated with both the rate of change of z and the external velocity v. However, during plastic motion, this formulation becomes problematic. The presence of the elastic term $\sigma_0 z$ introduces an offset force that counteracts the pure damping behavior expected in the plastic phase. As a result, instead of exhibiting a free, drift-like motion driven solely by the applied force, the system generates an undesired restoring force that biases the motion. Furthermore, when the external human-applied force $F_h \to 0$, the elastic term causes the internal state to pull the system back, producing an elastic rebound that contradicts the notion of a permanent plastic displacement. For these reasons, the elastic contribution must be effectively removed whenever the controller enters the plastic regime.

To achieve this, system (2.6) is modified as follows:

$$\begin{cases} F_r &= \sigma_0(z-w) + \sigma_1 \dot{z} + \sigma_2 v \\ \dot{z} &= v - \alpha(z)|v|\frac{\sigma_0}{f_c} z \\ \dot{w} &= \alpha(z)\frac{z-w}{\tau_w} \end{cases} \tag{2.10}$$

The key idea of this modified formulation is the introduction of an additional internal state variable, w, which acts as a plastic tracking state. The elastic force is now governed by the difference $(z-w)$ rather than by z alone. In the elastic regime, where $\alpha(z) = 0$, the variable w remains constant and the model behaves identically to the standard LuGre model: the restoring force $\sigma_0(z-w)$ reduces to $\sigma_0 z$, enabling reversible pre-sliding deformation.

When the system transitions into the plastic regime, i.e., when $\alpha(z) > 0$, the dynamics of w are activated through the third equation. In this phase, w gradually follows the value of z with a rate determined by the time constant τ_w. As w approaches z, the term $(z-w)$ decreases, progressively eliminating the elastic component from F_r. This mechanism ensures that the internal stress associated with the elastic deformation is released, allowing the robot to undergo a net, irreversible displacement while avoiding any spring-like restoring effects.

The parameter τ_w therefore governs how rapidly the controller removes the elastic component once plastic motion has been detected. A small value of τ_w results in fast

convergence, enabling almost immediate elimination of the elastic force, while larger values allow for a gradual transition into the plastic phase. Because the evolution of w is gated by $\alpha(z)$, the model ensures that plastic behavior is expressed only when the applied force exceeds the breakaway threshold, thereby preserving true elastic behavior below this threshold.

Figure 2.5 illustrates the resulting elasto–plastic behavior for different values of $\alpha(z)$. The top plot (Fig. 2.5b) shows the evolution of the internal states z and w: when $\alpha(z) = 0$, the two states diverge, preserving elastic deformation; when $\alpha(z) > 0$, w converges toward z, nullifying the elastic term. The bottom plot (Fig. 2.5c) shows the resulting Cartesian displacement of the system under three force steps. The first and third steps have equal amplitude, while the second is five times larger, producing a significant plastic deformation. The figure illustrates how the model captures both reversible elastic motion and irreversible plastic deformation, depending on the applied force.

2.3.3 Model Extension for 3D Motion

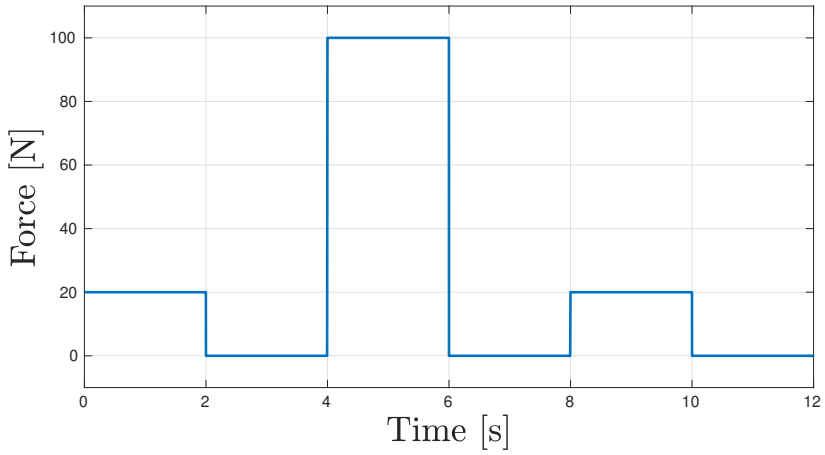
Using the elasto-plastic model as a Cartesian controller requires extending the single-axis formulation to three-dimensional space. The most direct approach is to apply the one-dimensional model independently along each of the Cartesian axes.

However, this extension fails to produce smooth spatial motion. Since each axis evolves according to its own friction dynamics without considering the others, the resulting 3D trajectory may exhibit discontinuities or undesirable axis-by-axis locking effects, especially when the direction of motion changes. This decoupled behavior is incompatible with the continuous and direction-consistent response expected from a Cartesian compliant controller.

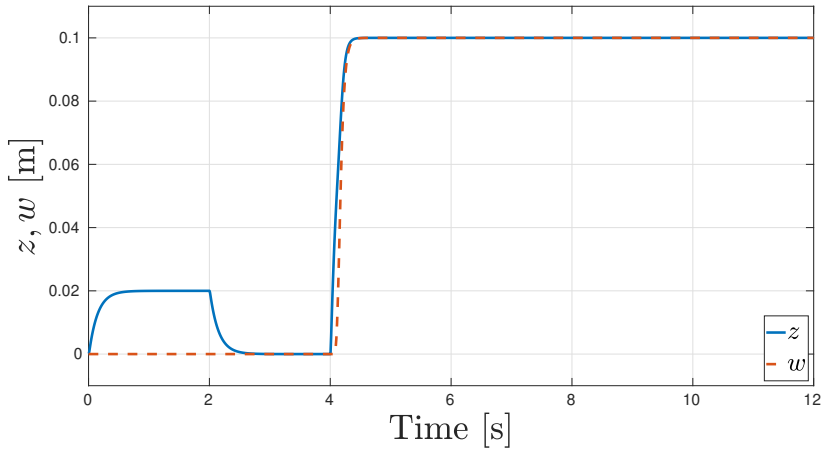
Specifically, per-axis elasto-plastic controllers treat each Cartesian direction independently and decide, along the x, y, and z axes, whether it is in an elastic or plastic state. When a force vector is applied in some arbitrary direction, each axis may:

- enter plastic motion at a different time,
- transition at a different rate,
- remains elastic while the others yield

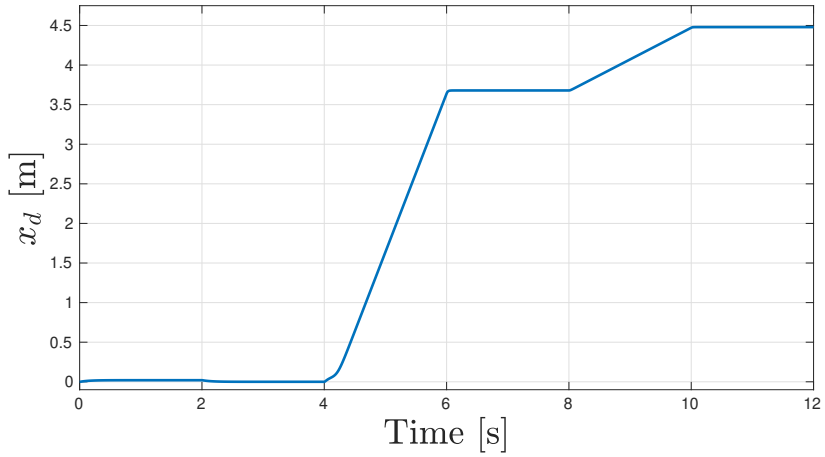
As a result, the actual Cartesian motion becomes axis-aligned and piecewise, producing trajectories with jerky transitions or "zig-zag" behavior. As human-applied forces are naturally vectorial, and not axis-separable, decoupled controllers feel artificial and non-smooth.



(a) Input force



(b) State variables z (blue line) and w (dashed red line).



(c) Cartesian displacement.

Figure 2.5 Elasto-plastic behavior. Three force steps are applied, from 0 s to 2 s, from 4 s to 6 s, and from 8 s to 10 s, where the first and third have the same amplitude, a fifth of the second. $z_{ba} = 0.03$ m, $z_{ss} = 0.1$ m

A more appropriate multidimensional formulation is proposed in [124] for the two-dimensional case, where a coupling term $\lambda(\mathbf{v})$ is introduced to coordinate the evolution of the internal states across axes. For a planar velocity vector $\mathbf{v} = [v_x, v_y]^\top$, the coupling term is defined as

$$\lambda(\mathbf{v}) = \frac{\|C_k^2 \mathbf{v}\|}{g_\mu(\mathbf{v})}, \tag{2.11}$$

where $g_\mu(\mathbf{v})$ is given by

$$g_\mu(\mathbf{v}) = \frac{\|C_k^2 \mathbf{v}\|}{\|C_k \mathbf{v}\|} + \tilde{g}_\mu(\mathbf{v}), \tag{2.12}$$

and

$$C_k = \begin{bmatrix} \mu_{kx} & 0 \\ 0 & \mu_{ky} \end{bmatrix}. \tag{2.13}$$

Here, μ_{ki} denotes the dynamic friction coefficient along axis i, and $\tilde{g}_\mu(\mathbf{v})$ represents the Stribeck component. The introduction of $\lambda(\mathbf{v})$ couples the friction dynamics across axes, ensuring that the transition from elastic to plastic behavior depends on the overall direction and magnitude of motion, rather than on each axis independently. This coupling is crucial to guarantee smooth multidimensional trajectories.

To extend this formulation to three dimensions, one defines $\mathbf{v} = [v_x, v_y, v_z]^\top$ and generalizes the friction matrix as $C_k = \mathrm{diag}(\mu_{kx}, \mu_{ky}, \mu_{kz})$. By removing the Stribeck term and substituting (2.12) into (2.11), the expression for the coupling term simplifies to

$$\lambda(\mathbf{v}) = \sqrt{(\mu_{kx} v_x)^2 + (\mu_{ky} v_y)^2 + (\mu_{kz} v_z)^2}, \tag{2.14}$$

which directly couples the Cartesian velocity components through a weighted norm induced by the friction coefficients.

With this coupling term, the multidimensional state dynamics become [124]

$$\dot{z}_i = v_i - \frac{\lambda(\mathbf{v})}{\mu_{ki}} \frac{\sigma_0}{f_c} z_i, \qquad i = x, y, z, \tag{2.15}$$

ensuring that the rate of internal deformation on each axis scales consistently with the overall direction of motion. In the application considered here, the desired behavior is isotropic, meaning that all axes should respond identically. Thus, the dynamic friction coefficients are chosen such that

$$\mu_k = \mu_{kx} = \mu_{ky} = \mu_{kz}. \tag{2.16}$$

Under this assumption, the friction weighting collapses to a uniform scalar, and the coupling term simplifies to the Euclidean norm of the Cartesian velocity:

$$\frac{\lambda(\mathbf{v})}{\mu_{ki}} = \|\mathbf{v}\|. \tag{2.17}$$

This isotropic formulation guarantees smooth, direction-consistent motion in 3D space while retaining the core elasto–plastic characteristics.

Finally, when integrating the elastic extension introduced in Section 2.3.1 and computing the switching function $\alpha(z)$, the scalar internal state z is replaced by the L^2-norm of the vector of internal states $\mathbf{z} = [z_x, z_y, z_z]^\top$. This ensures that the transition between elastic and plastic behavior is triggered based on the magnitude of the combined multidimensional internal deformation, in accordance with the model's isotropic nature.

2.4 Motion Reversal

The model also requires refinement to address the inversion of motion. For simplicity, consider the single-axis case. By combining Newton's law of motion, (2.1), with the modified LuGre elasto–plastic model (2.10), the controller dynamics can be expressed as

$$\begin{cases} \ddot{x} &= M^{-1}(F_h - \sigma_0(z-w) - \sigma_1\dot{z} - \sigma_2 v) \\ \dot{z} &= v - \alpha(z)|v|\frac{\sigma_0}{f_c}z \\ \dot{w} &= \alpha(z)\frac{z-w}{\tau_w} \end{cases} \tag{2.18}$$

Here, x represents the Cartesian position, F_h is the human-applied force, z and w are the internal states described previously, and $\alpha(z)$ determines the transition between elastic and plastic regimes.

Now, consider the scenario of plastic motion, with $v > 0$. If $\dot{z} \neq 0$, inverting the direction of v immediately changes the sign of $\dot{z}$. When the internal state z falls below the breakaway threshold z_{ba}, the function $\alpha(z)$ switches to zero. This immediately freezes the evolution of w ($\dot{w} = 0$), which has the unintended consequence of restoring the elastic term $\sigma_0(z-w)$ in the force equation. In other words, the system temporarily reintroduces a spring-like restoring force during plastic motion, resulting in small, undesired oscillations. Figure 2.6b illustrates this behavior.

This issue is not limited to the single-axis case. In three dimensions, the same phenomenon occurs whenever $\|\dot{\mathbf{z}}\|$ falls below z_{ba}. The elastic term is reactivated prematurely, resulting in a varying force that interferes with the desired plastic displacement. Once the

norm $\|\mathbf{z}\|$ begins increasing again, the system resumes plastic motion, but the temporary reintroduction of the elastic force can degrade the smoothness and predictability of the motion.

To address this problem, an auxiliary state r is introduced. This additional dimension effectively "locks" the model in its plastic state. Specifically, the original switching function $\alpha(\|\mathbf{z}\|)$ is replaced by a modified version $\alpha(\|\mathbf{z_e}\|)$, where

$$\mathbf{z_e} = [z_x, z_y, z_z, r]^\top$$

includes the extra state r alongside the internal states in each axis. The evolution of r is defined as:

$$\dot{r} = \dot{\alpha}(z), \tag{2.19}$$

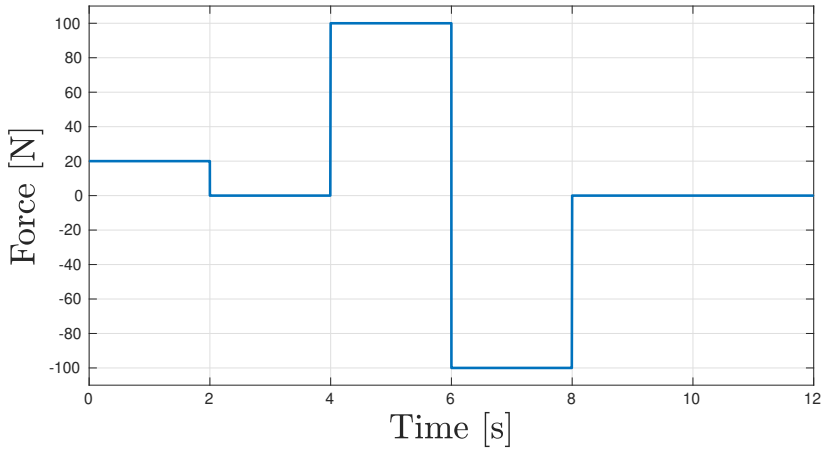
where $\dot{\alpha}(z)$ is nonzero only in the transition region $z_{ba} \leq \|\mathbf{z}\| < z_{ss}$ and ensures that r increases gradually as the system enters the plastic regime. Once r reaches its upper bound z_{ss}, it maintains the system in the plastic state, preventing the elastic term from being reintroduced prematurely. Figure 2.3 illustrates the shape of $\dot{\alpha}(z)$.

With this modification, the controller preserves smooth plastic motion even when the internal states approach the breakaway threshold. Figure 2.6 shows the comparison: without r, the system exhibits intermittent elastic rebounds, whereas with r, the plastic displacement is continuous and monotonic, providing a stable and predictable human-guided response.

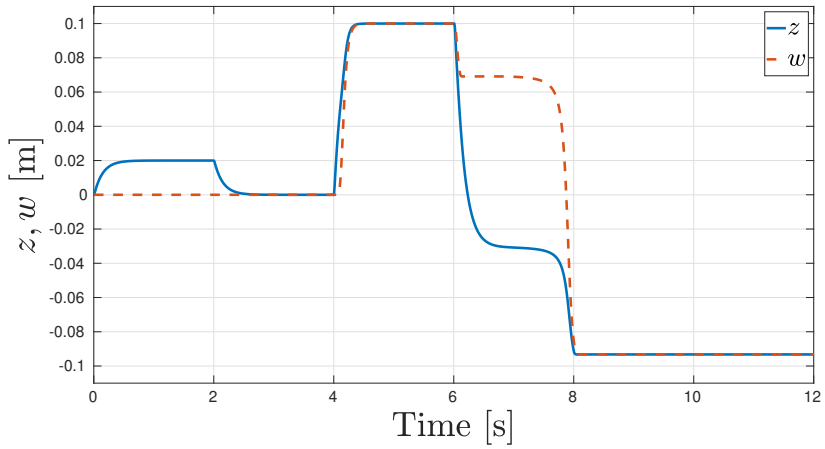
2.4.1 Reset Condition

The method introduced in the previous section successfully preserves the plastic motion once the system transitions into the plastic regime. This ensures that the elastic term $\sigma_0(z-w)$ is not prematurely reintroduced, allowing smooth and monotonic plastic displacement. However, this also introduces a limitation: the controller is effectively locked in the plastic state and cannot naturally transition back to the elastic regime. Without a mechanism to exit the plastic phase, the system would permanently disregard elastic behavior, which is undesirable for tasks requiring intermittent compliance or when the human operator reduces the applied force.

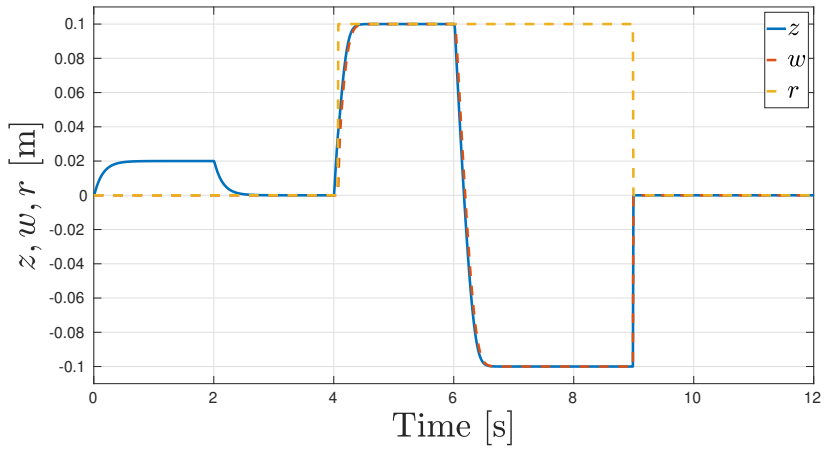
To address this, a plastic-to-elastic transition condition is defined based on the mechanical work exerted by the human operator on the robot. In this work, the internal states (z, w, r) are reset whenever the cumulative mechanical work over a short moving window falls below a



(a) Input force



(b) Evolution without r. The reversal of the input force brings back the system in its elastic state, blocking the evolution of w until $\alpha(z)$ starts growing again.



(c) Evolution with r. The plastic behavior is preserved until after the input force is removed (using reset condition (2.20)).

Figure 2.6 Comparison between the state variables z (blue line) and w (dashed red line) without and with the introduction of r (dashed yellow line).

predefined threshold. Formally, this condition is expressed as

$$\int_{t-T_r}^{t} \mathbf{F_h}^\top \mathbf{v}\, \mathrm{d}\tau < R_v, \tag{2.20}$$

where $\mathbf{F_h} = [F_{h,x}, F_{h,y}, F_{h,z}]^\top$ is the Cartesian force applied by the human operator, R_v is a positive constant representing the work threshold, and T_r is the duration of the moving window over which the work is evaluated. This integral effectively measures whether the human has significantly interacted with the system; when the applied work is small, the system is allowed to reset the internal states, restoring the possibility of elastic motion.

It is essential to note that this reset condition is only enabled when $\alpha(z) > 0$, i.e., when the system is already in or near the plastic regime. This ensures that the reset does not interfere with ongoing elastic behavior or cause spurious transitions. Figure 2.6c illustrates the effect of this reset mechanism, showing how the system can re-enter the elastic regime after the applied human work drops below the threshold, thereby enabling flexible and intuitive human-guided interactions while maintaining the benefits of the elasto-plastic model.

2.4.2 Complete Control Model

The resulting controller is the following:

$$\begin{cases} \ddot{\mathbf{x}} &= M^{-1}(\mathbf{F_h} - \sigma_0(\mathbf{z}-\mathbf{w}) - \sigma_1\dot{\mathbf{z}} - \sigma_2\mathbf{v}) \\ \dot{\mathbf{z}} &= \mathbf{v} - \alpha(\|\mathbf{z_e}\|)\|\mathbf{v}\|\frac{\mathbf{z}}{z_{ss}} \\ \dot{\mathbf{w}} &= \alpha(\|\mathbf{z_e}\|)\frac{\mathbf{z}-\mathbf{w}}{\tau_w} \\ \dot{r} &= \dot{\alpha}(\|\mathbf{z}\|) \end{cases} \tag{2.21}$$

with α computed using (2.7) and using the reset condition

$$\begin{cases} \mathbf{z} &= 0 \\ \mathbf{w} &= 0 \\ r &= 0 \end{cases} \text{, if } \alpha(\|\mathbf{z_e}\|) > 0 \wedge \int_{t-T_r}^{t} \mathbf{F_h}^\top \mathbf{v}\mathrm{d}\tau < R_v \tag{2.22}$$

where $\mathbf{z_e} = [z_x, z_y, z_z, r]^\top$ and $z_{ss} = f_c/\sigma_0$.

Table 2.3
EXPERIMENTS PARAMETERS

Parameters	Value	Unit
m_x, m_y, m_z	10	kg
σ_0	500	N/m
σ_1	50	Ns/m
σ_2	50	Ns/m
z_{ba}	0.1	m
z_{ss}	0.5	m
τ_w	0.02	s
T_r	1	s
R_v	0.1	J

2.5 Experiments

This controller has been implemented in C++ using the ROS1 framework [104] and the `cnr_ros_control` library [15]. The controller has been tested on the collaborative cell shown in Figure 2.7, which is equipped with a UR10e robot. Force data was collected using the internal force/torque sensor placed on the robot's wrist. The source code of the controller is available at [35]. Controller parameters are in Table 2.3.

In the first experiment (Figure 2.8), no trajectory is provided to the robot and three force impulses (a low value first, a high value then, and finally a low value again) are applied along the x-axis. The first impulse keeps the controller in its elastic state, holding $w = 0$, so that the displacement is canceled after the force is removed. On the contrary, the following impulse shifts the controller into a plastic state: the state variable w removes the elastic term, and the displacement operated in this state becomes permanent. Almost one second (the reset window size) after the force is removed, the system shifts to the elastic state again, as shown by the third impulse applied.

A second experiment (Figure 2.9) shows the capabilities of the controller in presence of a Cartesian trajectory. The robot makes a back-and-forth movement along the base y-axis. The operator is tasked with pushing and pulling along the x-axis to trigger both the elastic return and the trajectory deformation. It is worth noticing that the operator exerts forces also in other directions. During the forward movement, two small impulses are applied. The behavior obtained is the one expected from full mass-spring-damper admittance control. During the backward movement, the robot is dragged, thus shifting the controller into plastic state. The

trajectory is permanently deformed, and it can be seen that a constant delta between the robot Cartesian position and the trajectory is kept until the end of the trajectory. It can also be seen that the plastic force profile includes a small component on the y-axis. While not enough to produce a state shift of the controller, the actual plastic behavior allows it to produce a permanent motion even on the y-axis.

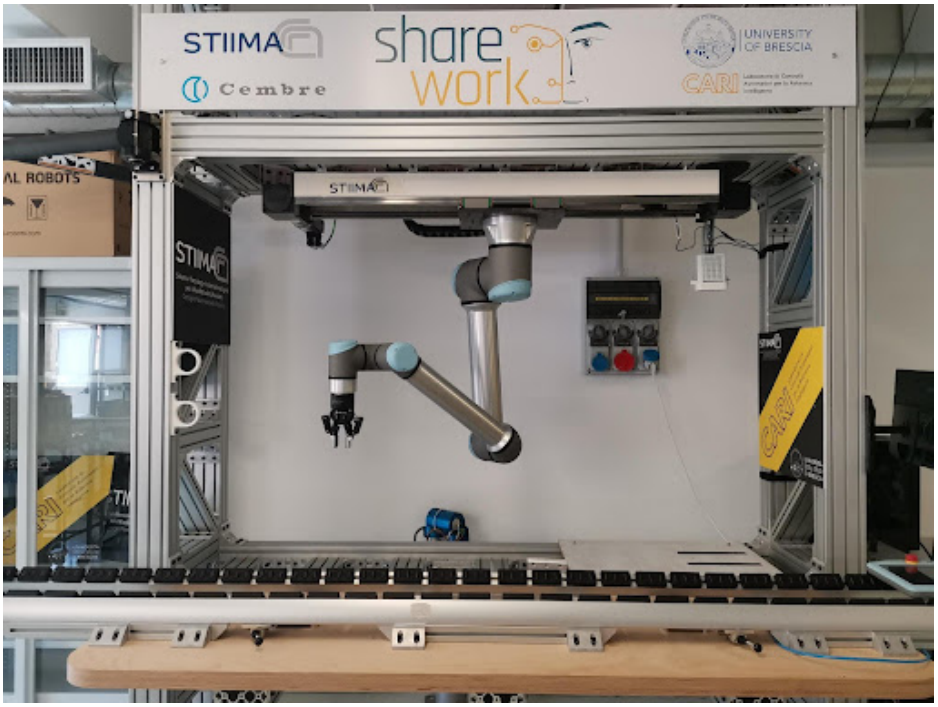
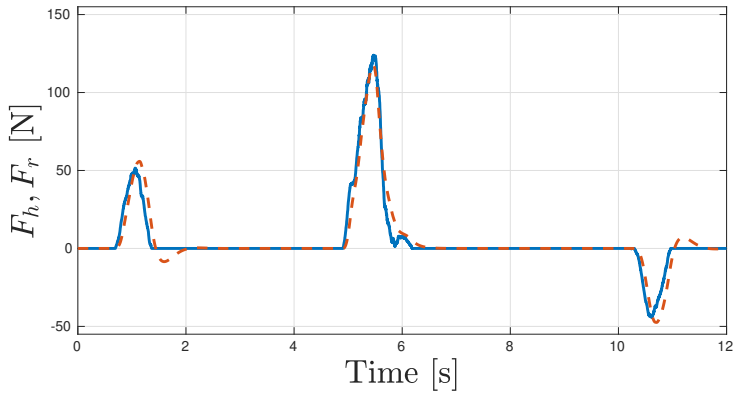
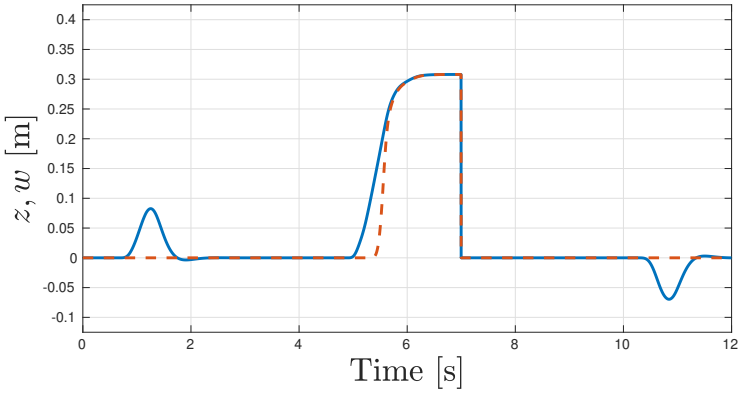


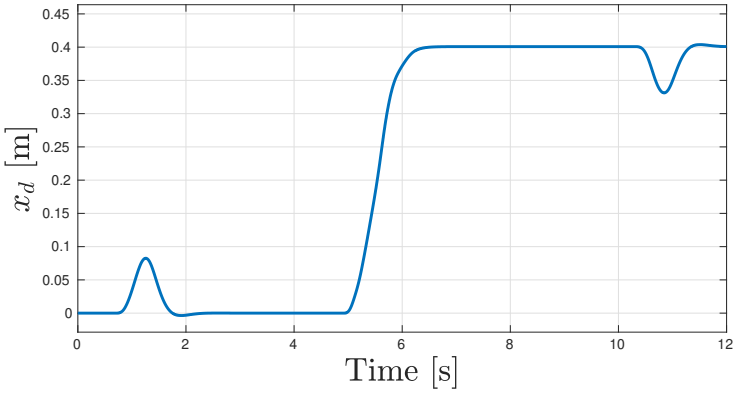
Figure 2.7 Collaborative cell. The 6-DoF UR10e robot includes a force-torque sensor on the wrist. The robot is mounted on a linear guide which, for the purpose of this work, was kept fixed.



(a) Force applied and resistant force computed by the controller. Applied force (blue line), resistant force (dashed red line).

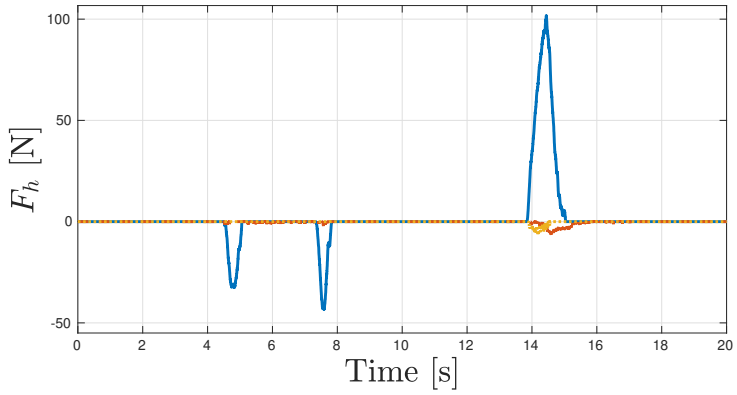


(b) State variables x-components. z state (blue line), w state (dashed red line)

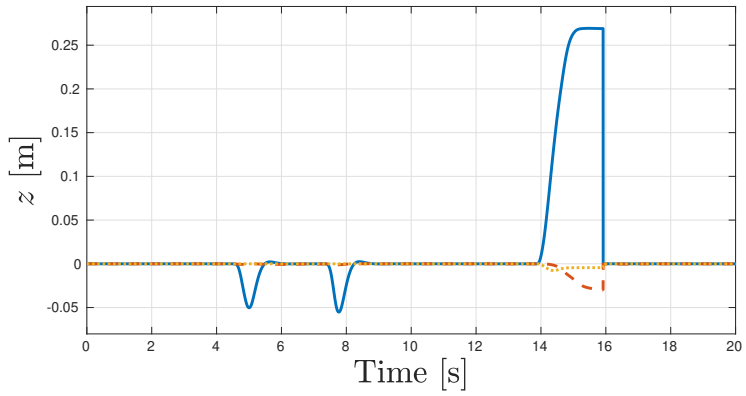


(c) Cartesian deviation from reference trajectory $x_r = 0$

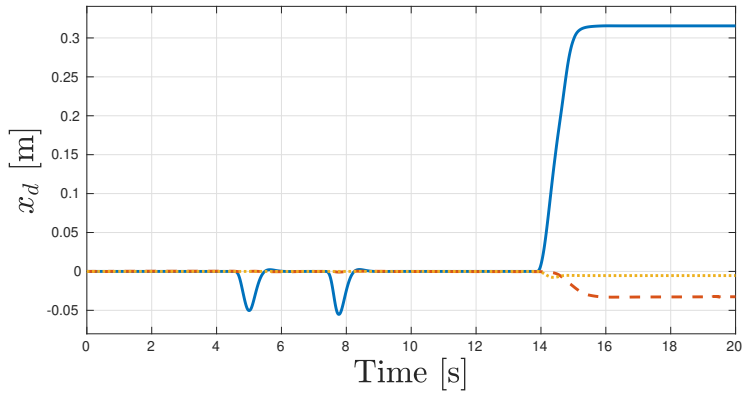
Figure 2.8 Deformation from steady-state condition. (a) Forces; (b) state variables; (c) trajectory deformation



(a) Applied Force



(b) State variables



(c) Cartesian deviation from reference trajectory x_r

Figure 2.9 Deformation during trajectory. x-component (blue line), y-component (dashed red line), z-component (dotted yellow line). (a) Applied force; (b) z state variables; (c) deformation of the trajectory (x_d) computed by the controller

2.6 Summary

This chapter introduces an elasto-plastic interaction model and its integration within an admittance control architecture for position-controlled manipulators, thereby overcoming the classical trade-off between ease of physical guidance and trajectory precision.

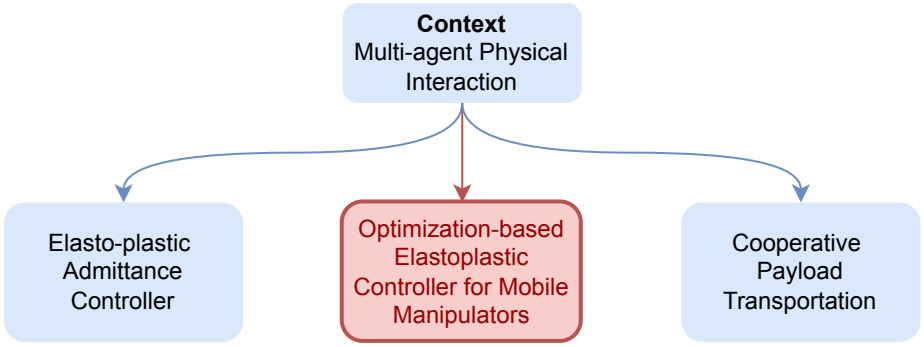
To achieve physically meaningful two-regime behavior, the controller utilized the LuGre friction model as a rheology-inspired internal dynamics and then introduced a set of modifications to make it suitable for physical human-robot interaction (pHRI) control. First, the LuGre state dynamics were augmented to explicitly reproduce actual elastic behavior below a breakaway displacement by enforcing stiction through a smooth switching function.
To prevent undesired elastic restoring effects during plastic motion, an additional internal state was introduced to progressively suppress the elastic component, enabling permanent offsets without rebound. The model was also extended to 3D motion in an isotropic and direction-consistent manner by coupling the axis dynamics through the velocity norm and by triggering the elastic-plastic transition based on the norm of the state variable $\mathbf{z}$, rather than through independent per-axis thresholds.

A dedicated treatment of motion reversal was then presented. Without further measures, reversing the direction of motion during plastic behavior can temporarily reintroduce elastic dynamics, resulting in unwanted oscillations and reduced predictability. The chapter addressed this by introducing an auxiliary state that prevents premature returns to the elastic regime while plastic motion is ongoing. Since this "plastic lock" must still allow the system to recover elastic behavior when interaction ceases, a reset mechanism based on the mechanical work of the applied wrench over a moving window was defined. This yields a complete control model that preserves smooth plastic evolution during reversal and reliably returns to the elastic regime when the interaction becomes negligible.

Finally, the proposed controller was implemented in ROS 1 and experimentally validated on a UR10e robot equipped with a wrist force/torque sensor. The results confirmed the intended qualitative behavior.

Chapter 3

Elastoplastic Control for Mobile Manipulation



This chapter presents an extension of elasto-plastic interaction control to whole-body mobile manipulation. The objective is to enable a mobile manipulator to respond to physical interaction with a dual behavior: small perturbations generate reversible, elastic deformations around a nominal reference, while sustained or sufficiently strong interaction produces persistent, plastic offsets that allow the robot to be guided away from the nominal path. The method is implemented within a hierarchical quadratic programming framework, allowing interaction objectives, kinematic feasibility, and secondary behaviors to be handled coherently under constraints. An arbitration strategy redistributes motion between the manipulator and the mobile base according to the current interaction regime, improving responsiveness in compliant phases and exploiting the base workspace when large deviations are required. A trajectory recovery mechanism is then introduced to support controlled re-entry toward the nominal path after a plastic deviation, by shaping an intermediate Cartesian reference within the optimization. The proposed controller is experimentally validated in a collaborative

transport scenario with a human partner, and its behavior is analyzed through tracking performance, regime transitions, and recovery performance, including a comparison with a multi-modal whole-body controller. Practical tuning considerations and implementation aspects are discussed.

The Chapter is structured as follows:

- **Section 3.1** motivates whole-body compliance for mobile manipulators in contact-rich and human-guided tasks, clarifying the chapter goals. It states the main contributions and working assumptions for the considered robotic platform and sensing/control stack.

- **Section 3.2** establishes the modeling and control background used throughout the chapter by describing the mobile manipulator kinematic structure and by introducing hierarchical quadratic programming as the optimization tool to manage task priorities and constraints in a stack-of-tasks formulation.

- **Section 3.3** defines the elasto-plastic interaction concept as adapted to mobile manipulation, including how elastic and plastic responses are produced, how motion reversals are handled to avoid undesirable transient behavior during direction changes, and how a reset strategy enables a return to elastic behavior after interaction ceases.

- **Section 3.4** presents the optimization-based elastoplastic controller, which embeds the interaction behavior into a hierarchical program. It also introduces a state-dependent arbitration policy that biases the solution toward arm motion in elastic phases and toward base motion in plastic phases, while remaining compatible with actuator and kinematic limits.

- **Section 3.5** introduces a trajectory recovery strategy that augments the optimization problem with an intermediate Cartesian reference, allowing the controller to manage plastic offsets without permanently sacrificing convergence to the nominal trajectory. The section explains how recovery is activated after the interaction state is restored and how this reference-level approach enables additional shaping objectives if required.

- **Section 3.6** describes the experimental methodology, including the collaborative transport setup, the limitations imposed by hardware dynamics and safety considerations, and the assessment approach used to evaluate tracking, interaction response, recovery behavior, and benchmarking against a reference whole-body controller.

- **Section 3.7** reports experimental results for trajectory tracking with elastic response, plastic deformation under strong interaction, and recovery to the nominal path. It then compares observed behaviors with the benchmark controller, highlighting qualitative and practical differences in regime transitions and re-entry behavior, and concludes with a focused discussion on parameter tuning sensitivity and deployment considerations for real mobile manipulation systems.
- **Section 3.8** concludes the Chapter and summarizes methodology and results presented

3.1 Introduction

3.1.1 Context

Mobile manipulation lies at the intersection of two traditionally distinct areas of robotics: mobile robotics and robotic manipulation. Classical industrial manipulators are typically mounted on fixed bases and operate within highly structured, well-calibrated environments such as assembly lines, where workspaces are carefully engineered, and sources of uncertainty are minimized. In contrast, mobile robots are primarily designed for navigation, localization, and environment sensing across large and often unstructured spaces. Still, they generally possess limited or no manipulation capabilities beyond simple payload transport. Mobile manipulators integrate these two paradigms by combining a mobile base with one or more articulated robotic arms, along with rich sensor suites for perception, localization, and interaction. This fusion enables robots to physically interact with a wide variety of objects over extended workspaces and in dynamically changing environments, making them well-suited for tasks such as large-scale machining, on-site assembly, warehouse logistics, and collaborative payload transport with humans or other robots.

Many of these applications inherently involve sustained physical interaction with the environment or with human partners, which requires compliant and safe behaviors to handle uncertainties and interaction forces robustly. Such compliance is commonly achieved through impedance or admittance control strategies, which regulate the dynamic relationship between motion and force. In the context of mobile manipulators, compliance can be implemented in two principal ways: either by applying the interaction control exclusively to the manipulator arm while treating the mobile base as a rigid, position-controlled platform, or by modeling and controlling the entire robot—including both the base and the arm—as a unified dynamical system. The former approach benefits from reduced modeling complexity and more straightforward implementation, as it leverages well-established control schemes developed

for fixed-base manipulators. However, it often neglects the dynamic coupling between the base and the arm, which can limit performance during forceful interactions or large motions. In contrast, whole-body or unified control approaches explicitly account for these couplings, enabling more coordinated behaviors. This allows the robot to distribute interaction forces across all degrees of freedom, resulting in smoother motion, improved balance, and enhanced safety during complex physical interactions.

3.1.2 Contribution

The main contribution consists of the deployment of a whole-body controller for mobile manipulators designed for physical human–robot interaction. The controller exhibits two distinct behaviors: an elastic behavior, which produces temporary deformations of a nominal reference trajectory, and a plastic behavior, which generates a permanent offset of that trajectory. The controller is implemented through a hierarchical quadratic programming framework that automatically prioritizes the use of either the base or the arm depending on the current elastic/plastic behavior. A trajectory recovery strategy is included to enable the robot, when necessary, to return to its original trajectory after a plastic offset has been introduced. The controller is validated in a collaborative payload transport scenario.

3.1.3 Assumptions

Throughout this work, the following assumptions are made:

- The mobile manipulator has a planar base.
- The mobile base is holonomic and is velocity-controlled (linear velocities along the two planar axes plus angular velocity around the normal axis).
- The arm manipulator is position-controlled, with sufficiently high bandwidth to track reference trajectories accurately.
- The arm is equipped with a force–torque sensor that provides reliable measurements of the contact forces and moments at the wrist.

3.2 Fundamentals

3.2.1 Mobile Manipulator

The system considered here is a mobile manipulator with an omnidirectional planar base. The mobile base kinematics is represented as a 3-DoF manipulator between the world frame W and the base frame B, consisting of two prismatic joints followed by a revolute joint. The configuration vector $q(t) \in \mathbb{R}^{3+n}$ of the full robot is

$$q(t) = [q_B^T \quad q_M^T] = [x_B, y_B, \theta_B, q_1, \ldots, q_n]^T \tag{3.1}$$

where $q_B = [x_B, y_B, \theta_B]^T$ are the planar position and the heading of the mobile base and $q_M = [q_1, \ldots, q_n]^T$ are the joint values of the manipulator. The kinematics equations of the mobile manipulators are

$$v_{ee} = J(q)\dot{q} \qquad a_{ee} = \dot{J}(q)\dot{q} + J(q)\ddot{q} \tag{3.2}$$

where v_{ee}, a_{ee} are the velocity and the acceleration twists of the end effector, $\dot{q}, \ddot{q}$ are the time derivatives of the configuration vector and $J(q) = [J_B(q_B) \;\; J_M(q_M)]$ is the manipulator Jacobian with respect to the world frame, with

$$J_B(q_B) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad J_M(q_B, q_M) = \begin{bmatrix} {}^W E_B(\theta_B) & 0 \\ 0 & {}^W E_B(\theta_B) \end{bmatrix} {}^B J_M(q_M) \tag{3.3}$$

${}^B J_m(q_m)$ is the Jacobian of the manipulator expressed in the manipulator base and ${}^W E_B(\theta_B)$ is the rotation matrix between base frame and world frame. Time dependencies are not indicated to simplify the notation.
At the acceleration level, the differential inverse kinematics can be solved analytically as

$$\ddot{q} = J_H^{\dagger}(a_{ee} - \dot{J}(q)\dot{q}) + (\mathbb{I} - J_H^{\dagger} J)\ddot{\xi} \tag{3.4}$$

with $J_H^{\dagger} = H^{-1}J^T(JH^{-1}J^T)^{-1}$ is the weighted pseudo-inverse of the Jacobian and $\ddot{\xi} \in \mathbb{R}^{3+n}$ secondary task vector projected in the nullspace of the Jacobian. The weight matrix H is

a symmetric positive semi-definite matrix used to weight differently each joint, in order to provide preferences on which ones should move. Another approach to the differential inverse kinematics problem is through optimal control. The differential kinematics equation is expressed as an optimization problem:

$$\min_{\ddot{q}} \|a_{ee} - \dot{J}(q)\dot{q} - J(q)\ddot{q}\|^2 \tag{3.5}$$

This formulation allows for integration of constraints and the extension to hierarchical quadratic programs.

By modeling the joint as a double integrator and by means of backward Euler discretization, the optimization problem can be rewritten as

$$\min_{\ddot{q}} \|a_{ee} - \dot{J}(q_k)\dot{q}_k - J(q_k)\ddot{q}\|^2 \tag{3.6}$$

where $q_k = q(k\Delta t)$ and Δt is the sampling time.

Due to the numeric nature of the discrete integration, every step introduces an error in the overall solution that, if left untreated, can generate a huge drift in the computation of the inverse kinematics. The standard solution to this problem is to add terms to the Cartesian acceleration that are proportional to the task position error and the task velocity error. The new formulation of the minimization problem, called Closed Loop Inverse Kinematics (CLIK), is then

$$\min_{\ddot{q}} \quad \|a_{ee,r} - \dot{J}(q_k)\dot{q}_k - J(q_k)\ddot{q} + K_p(x_{ee,r} - x_{ee}) + K_v(v_{ee,r} - v_{ee})\|^2 \tag{3.7}$$

with $K_p, K_v \in \mathbb{R}^{6\times 6}$ and $v_{ee,r}, x_{ee,r} \in \mathbb{R}^6$ the reference position and velocity.

3.2.2 Hierarchical Quadratic Programming

The mobile manipulator, as a unique system, can be considered a redundant manipulator. As such, the degrees of redundancy can be exploited to perform secondary tasks in the null space. A standard method to achieve this is by using a Stack of Tasks (SoT), where each task to perform is associated with a level of importance. The Stack, therefore, defines a hierarchy of tasks, from the most to least important. The higher the level in the stack, the more accurate the solution of the relative task will be. Optimization problems defined with SoT can be divided in two categories: strict hierarchy and soft hierarchy. In a strict problem, the hierarchy defined in the SoT is solved in a lexicographic manner, i.e., the best possible

solution is found for a task before considering the lower level ones. Mathematically, a series of QP problems is solved sequentially, where the first is

$$\begin{aligned} \min_{x} \quad & \|A_0 x + b_0\|^2 \\ & Cx + d = 0 \\ & Ex + f \geq 0 \end{aligned} \tag{3.8}$$

with x_0^* the optimal solution. The subsequent problems are

$$\begin{aligned} \min_{x} \quad & \|A_k x + b_k\|^2 \\ s.t. \quad & A_0 x = A_0 x_0^* \\ & \vdots \\ & A_{k-1} x = A_{k-1} x_{k-1}^* \\ & Cx + d = 0 \\ & Ex + f \geq 0 \end{aligned} \tag{3.9}$$

for $k > 0$, where $A_0, \ldots, A_{k-1} \in \mathbb{R}^{m \times n}$ and $b_0, \ldots, b_{k-1} \in \mathbb{R}^m$ are the $k-1$ tasks matrices and $x_0^*, \ldots, x_{k-1}^* \in \mathbb{R}^n$ are the optimal solutions of the previous tasks. The k-th task is then solved by constraining the solution to be in the null-space of the previous $(k-1)$-th A task matrices. The equality constraints are expressed through the $C \in \mathbb{R}^{p \times n}$ matrix and $d \in \mathbb{R}^p$ vector, while the inequality constraints through $E \in \mathbb{R}^{s \times n}$ matrix and $f \in \mathbb{R}^s$ vector. Instead, in the soft hierarchical approach, a unique optimization problem is formalized that integrate all the tasks in a single cost function. Using this formalization, higher priority tasks can be relaxed to improve the solution of lower priority ones:

$$\begin{aligned} \min_{x} \quad & \lambda_0 \|A_0 x + b_0\|^2 + \cdots + \lambda_k \|A_k x + b_k\|^2 \\ s.t. \quad & Cx + d = 0 \\ & Ex + f \geq 0 \end{aligned} \tag{3.10}$$

with $\lambda_0, \cdots, \lambda_k \in \mathbb{R}$, $\lambda_0 \geq \cdots \geq \lambda_k$ the task weights.

3.3 Elastoplastic Formulation for Mobile Manipulation

In Chapter 2, it was demonstrated that a control algorithm suitable for physical human-robot interaction can be developed by starting from a rheological model, such as the LuGre friction model. The model, which portrays the interaction between two bodies as a bending bristle,

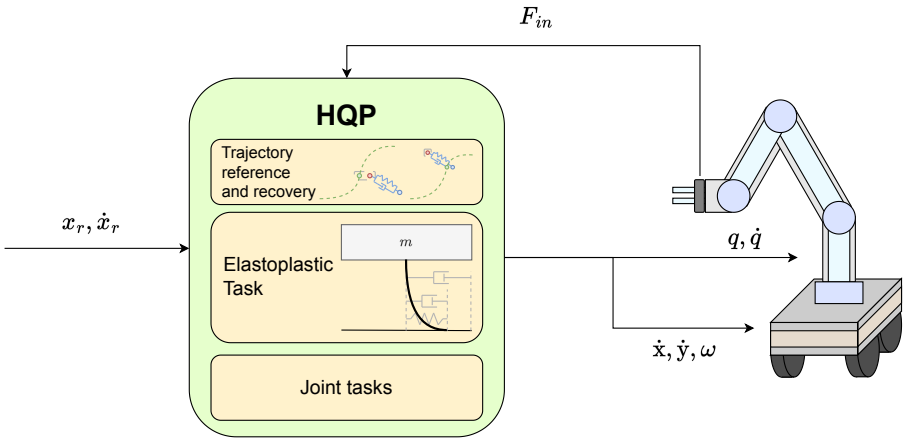


Figure 3.1 Visual Structure of the HQP elastoplastic controller for mobile manipulators

build the friction reaction to relative movement from the sum of two physical behaviors: the bristle bending generates an elastic force that restore the initial state of the bristle ones the exciting force ceases; the sliding of the bristle produces a permanent plastic offset in the relative position between the interacting surfaces. Here, an alternative formulation for the elastoplastic algorithm is provided, founded on the LuGre model.

Problem	**Description**
Lack of static friction	Forces below defined threshold (fixed or velocity-dependent) produce permanent displacement
Mono-dimensionality	The original version of the LuGre model is defined for the mono-dimensional case. Trivially applying the model to each dimension results in dimension-dependent behavior
Motion reversal	Quickly inverting the motion direction while performing major plastic deformations leads the model to switch to the elastic behavior before plastically moving along the new path
Plastic displacement	High forces still produce elastic deformation due to the presence of the stiffness term in the force equation
Reset condition	If the $\sigma_0 z$ elastic term is removed, an additional reset condition is necessary to bring the value of z back to zero if no force is provided

Table 3.1 Required LuGre model adjustments to obtain a model suitable for admittance control

3.3.1 Elastic and Plastic Displacement

As detailed in 2.3.1, the LuGre model can be extended to include static friction, i.e., the property of a friction model for which forces below the breakaway threshold do not produce permanent motion between the interacting forces. Alongside pre-sliding displacement, i.e., the reversible motion produced for forces below the breakaway threshold, these properties grant the LuGre model a range of forces for which its behavior is purely elastic. The extended model under consideration defines a sigmoid $\alpha(z)$, a function of the internal state of the model, that modulates the non-elastic term in the state equation of the model (2.6).

To obtain an elastoplastic behavior, that is, a continuous behavior in which elastic and plastic motion are the two extremes, the inclusion of stiction is not enough, as an elastic component is still present after the breakaway threshold. By exploiting the function $\alpha(z)$ again, it is possible to modulate the stiffness as a function of the state and eventually break the elasticity of the motion when required. This is obtained by replacing the fixed stiffness parameter σ_0 with a variable stiffness σ_α in the dynamic equation, defined as

$$\sigma_\alpha = \sigma_0(1-\alpha(z)) \tag{3.11}$$

The new model is then

$$\begin{cases} F_r &= \sigma_0(1-\alpha(z))z+\sigma_1\dot{z}+\sigma_2 v \\ \dot{z} &= v-\alpha(z)|v|\frac{\sigma_0}{f_c}z \end{cases} \tag{3.12}$$

and $\alpha(z)$ defined in (2.7).
For $|z| < z_{ba}$, $\sigma_\alpha = \sigma_0$ and the model behavior is equivalent to a spring-damper, where every movement is compensated once the external force applied ceases its action. On the contrary, $|z| \geq z_{ss}$ brings $\sigma_\alpha = 0$, for which the friction force of the model is generated by motion alone and not by position offset, which is free to change consequence-free.

3.3.2 Motion Reversal

As described in more detail in 2.4, inverting the direction of a force provided to the system results in the state variable z moving towards zero and then growing again with the opposite sign. This transition is appropriate when an elastic effect is still present in the response of the system, but it can create an unwanted shift in the behavior of the controller when $|z| \geq z_{ss}$. Especially in human-robot interaction scenarios, the transition from a non-elastic response to an elastic one is usually not anticipated by the human operator.

Since this shift is dictated by the state equation of the model, by setting the state derivative to zero once $\alpha(z) = 1$, the system state is prevented from decreasing, thereby eliminating this inversion effect.

$$\dot{z} = \begin{cases} 0 & sign(z) \neq sign(\dot{x}) \ \wedge \ \alpha(z) = 1 \\ \dot{x} - \alpha(z)\frac{z}{z_{ss}}\|\dot{x}\| & otherwise \end{cases} \tag{3.13}$$

3.3.3 Reset Condition

By removing the elastic term and preventing the evolution of the state variable z during motion inversion, when $\alpha(z) = 1$, the system ensures that a permanent change in position occurs, provided only by the force applied to it. To recover the dynamics of z and restore the elastic effect of the system, the algorithm requires a transition strategy. As described in more detail in 2.20, a reset condition based on the mechanical work of the contact force is defined

$$\int_{t-w_h}^{t} \mathbf{F_h}^{\top} \mathbf{v} \, \mathrm{d}\tau < R_v, \tag{3.14}$$

with R_v the work threshold and w_h a mo When the condition is satisfied, for $\alpha(z) = 1$, the internal state z is set to zero.

3.4 Optimization-based Elastoplastic Controller

To enable the use of the elastoplastic algorithm along multiple axes, the strategy presented in 2.3.3 is used. So, instead of using a scalar z, a vector $\mathbf{z} \in \mathbb{R}^j$ is used to represent the displacement along each of the j-axis considered. Also, the absolute value of z is replaced with the $\mathscr{L}_2$-norm $\|\mathbf{z}\|$.

To apply the elasto-plastic algorithm on a mobile manipulator, a lexicographic quadratic program (HQP) is defined. The optimization problem encompasses both the admittance behavior in Cartesian space and the robot's kinematics, as well as additional tasks that depend on the degree of redundancy of the robot. The problem is defined as follows

$$\begin{aligned} \underset{\ddot{q}}{\text{lex min}} \quad & \{T_1(\ddot{q}), \ldots, T_n(\ddot{q})\} \\ \text{s.t.} \quad & \ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max} \\ & \dot{q}_{\min} \leq \dot{q}_{t-1} + \ddot{q}\Delta t \leq \dot{q}_{\max} \\ & q_{\min} \leq q_{t-1} + \dot{q}_{t-1}\Delta t + \tfrac{1}{2}\ddot{q}\Delta t^2 \leq q_{\max} \end{aligned} \tag{3.15}$$

where $\{T_1(\ddot{q}), \ldots, T_n(\ddot{q})\}$ is the hierarchy of tasks to solve, and constraints on the argument and on the joint limits are put in place to limit avoid unfeasible commands to the hardware.

The higher-level task is used to enforce the elastoplastic dynamic. Considering the admittance equation

$$M(\ddot{x} - \ddot{x}_r) + \sigma_2(\dot{x} - \dot{x}_r) + \sigma_1 \dot{z} + \sigma_0 z = F \tag{3.16}$$

with $x_r, \dot{x}_r, \ddot{x}_r$ the reference trajectory, it can be rewritten as follow

$$\ddot{x} = \ddot{x}_r - M^{-1}\sigma_2(\dot{x} - \dot{x}_r) - M^{-1}\sigma_1 \dot{z} - M^{-1}\sigma_0 z + M^{-1}F \tag{3.17}$$

$\ddot{x}$ is the desired Cartesian acceleration generated by the dynamics of the admittance. Thus, by including (3.17) in the kinematics used in task (3.5)

$$\begin{array}{rcl} T_1(\ddot{q}) & = & \|J\ddot{q} + \dot{J}\dot{q} - \ddot{x}_r + M^{-1}\sigma_2(\dot{x}_r - \dot{x}) + M^{-1}\sigma_1 \dot{z} + M^{-1}\sigma_0 z - M^{-1}F\|^2 \\ & = & \|J\ddot{q} + \dot{J}\dot{q} - \ddot{x}_r + K_{v1}(\dot{x}_r - \dot{x}) + K_{v2}\dot{z} + K_p z - \ddot{x}_F\|^2 \end{array} \tag{3.18}$$

When the model renders elastic behavior, this is equivalent to a second-order CLIK task minus a contact force-dependent term $\ddot{x}_F$. The second task minimize the joint velocity error w.r.t. a reference $\dot{q}_r$:

$$T_2(\ddot{q}) = \|\dot{q} - \dot{q}_r\|_W^2 \tag{3.19}$$

while the third one is used for regularization

$$T_3(\ddot{q}) = \|\ddot{q}\|_W^2 \tag{3.20}$$

Both of these tasks accept a weighting matrix W.

3.4.1 Arbitration Policy for Null-space Optimization

By solving the inverse kinematics for the entire system in one step, the joints of the manipulator and the virtual joints of the mobile base are considered equal in terms of importance, ensuring that the solution accounts for movements of all the joints. To maximize the benefits of the different behaviors enabled by the elastoplastic algorithm, the motion of various robot parts can be prioritized. In this work, the following responses of the robot, related to the elastoplastic behaviors, are sought:

- If the elastoplastic controller is producing an elastic response, the movement of the sole arm of the robot is privileged.

- If the elastoplastic controller is producing a plastic response, the movement of the base is encouraged.

By taking into account the internal structure of the elastoplastic algorithm, this distribution results in have higher bandwidth and thus faster responses. By contrast, large offsets from the original trajectory are handled by the base, exploiting its large workspace. This difference in response is included in the HQP by means of the following state-dependent weighting matrix for tasks T_2 and T_3

$$W_z = W(z(t)) = \begin{bmatrix} W_{base} & 0 \\ 0 & W_{manipulator} \end{bmatrix} \tag{3.21}$$

with $W_{manipulator} = \mathbb{I}_{n\times n}$, $W_{base} = \frac{c}{1+k_\alpha \alpha(z)}\ \mathbb{I}_{3\times 3}$ and $k_\alpha \gg c \gg 1$. The relative weights between joints of the manipulator and virtual joints of the base are now dependent on $\alpha(z)$ and thus on the actual state of the elastoplastic algorithm. One of the main limitations of mobile robots is their limited velocities and accelerations, especially for indoor and industrial platforms. These limits are embedded in the optimization problem as constraints, but from a user's point of view, they can limit the actual response of the robot when plastic behavior is in use. By influencing only the cost function of the optimization, the prioritization of the arm versus the base must be compliant with the problem's constraints; otherwise, it is overridden. For example, in cases where the mobile base is moving at maximum speed or maximum acceleration for plastic motion, displacements of the arm can be implemented through optimization to successfully accomplish tasks that could not be completed solely by the base kinematics. Consequently, the arbitration policy should be interpreted as a preference that improves responsiveness and task continuity when feasible, while gracefully degrading to alternative joint usage under constraints.

3.5 Trajectory Recovery

After a phase in which the elastoplastic algorithm produces a plastic response, a permanent offset is added to the reference trajectory, resulting, by the end of the path, to a final position different from the one predefined at the start (Figure 3.2). There are instances where this approach is undesirable. For instance, allowing the operator to locally deviate from a precomputed trajectory (e.g., to account for previously unknown obstacles or to provide manual guidance on specific segments) is useful, but after the intervention, the controller should safely converge back to the nominal path. To handle these scenarios, a variant of the optimization-based elastoplastic controller is introduced. Taking inspiration from the work

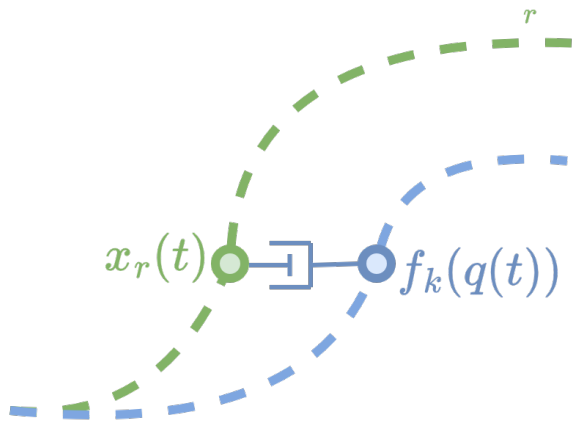


Figure 3.2 Plastic deformation of a trajectory r

in [120], the HQP in Sec. 3.4 is extended by including, in addition to the joint accelerations, the Cartesian acceleration of an intermediate (optimized) reference $x_d(t)$. The resulting lexicographic problem is

$$\begin{aligned} \underset{\ddot{q},\ddot{x}_d}{\text{lex min}} \quad & \{T_1(\ddot{q},\ddot{x}_d),\ldots,T_m(\ddot{q},\ddot{x}_d)\} \\ s.t. \quad & \ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max} \\ & \ddot{x}_{d,\min} \leq \ddot{x}_d \leq \ddot{x}_{d,\max} \\ & \dot{q}_{\min} \leq \dot{q}_{t-1} + \ddot{q}\Delta t \leq \dot{q}_{\max} \\ & \dot{x}_{d,\min} \leq \dot{x}_{d,t-1} + \ddot{x}_d\Delta t \leq \dot{x}_{d,\max} \\ & q_{\min} \leq q_{t-1} + \dot{q}_{t-1}\Delta t + \tfrac{1}{2}\ddot{q}\Delta t^2 \leq q_{\max} \\ & x_{\min} \leq x_{t-1} + \dot{x}_{d,t-1}\Delta t + \tfrac{1}{2}\ddot{x}_d\Delta t^2 \leq x_{\max} \end{aligned} \tag{3.22}$$

Here, $\ddot{x}_d \in \mathbb{R}^6$ denotes the acceleration of the desired intermediate Cartesian reference $x_d(t)$. Constraints related to the Cartesian limits of the robot can be included to prevent non-trackable references. The HQP task stack is modified as follows. The highest priority task is

$$T_1(\ddot{q},\ddot{x}) = \|\dot{x}_d(t) - \dot{x}_r(t)\|^2 \tag{3.23}$$

which enforce $x_d(t)$ to follow the nominal trajectory $\dot{x}_r(t)$ by tracking its velocity. The second task is

$$T_2(\ddot{q}) = \|J\ddot{q} + \dot{J}\dot{q} - \ddot{x}_d + M^{-1}\sigma_2(\dot{x}_d - \dot{x}) + M^{-1}\sigma_1\dot{z} + M^{-1}\sigma_0 z - M^{-1}F\|^2 \tag{3.24}$$

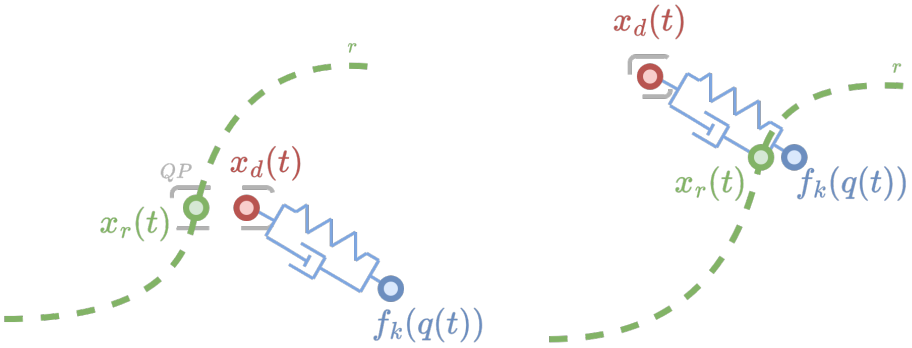


Figure 3.3 Recovery of the original trajectory. An intermediate Cartesian reference $x_d(t)$ is computed and moved such that the actual position of the end effector $f_k(q(t))$ returns on the precomputed path.

which computes the elastoplastic admittance/CLIK response with respect to the optimized reference $x_d(t)$, rather than directly with respect to the nominal reference $x_r(t)$ (as in (3.18)). The last two tasks are the same defined in (3.19) and (3.20).

During the elastic motion, and transitions from elastic to plastic, the desired reference match the trajectory reference as there are no higher priority task that influence $\ddot{x}_d$ and task (3.23) fully determine the evolution of $x_d(t)$ (constraints permitting), leaving no null space for lower-priority tasks to alter the intermediate reference. After a plastic motion is completed, having produced the non-temporary offset $x(t) - x_d(t) = \Delta x_p$, and after the elastoplastic algorithm is restored to the elastic state, a new task is placed at the top of the stack

$$T_{rec}(\ddot{q}, \ddot{x}) = \|x_d(t) - (x_r(t) - \Delta x_p)\|^2 \tag{3.25}$$

This task drives the intermediate reference toward a shifted version of the nominal trajectory. As a result, $x_d(t)$ compensates the plastic offset, while the subsequent admittance task drives the end-effector to follow $x_d(t)$, enabling the controller to converge back to the original trajectory $x_r(t)$.

Formulating the path recovery through a Cartesian reference in the optimization, additional purely Cartesian tasks can be included to shape the convergence to the reference trajectory, e.g., obstacle avoidance, speed and direction of convergence, etc.

Figure 3.4 Tiago Pro used for the experiments, mounting Robotiq force-torque sensor and gripper

3.6 Design of Experiments

3.6.1 Experimental Setup Description

To verify the response of the control strategy proposed in this chapter, the optimization-based elastoplastic controller has been tested in a collaborative payload transport scenario between a human operator and a mobile manipulator. The robot used is the Tiago Pro [116] by PAL Robotics, a two-armed mobile manipulator with 7 degrees of freedom for the arm, plus 1 degree of freedom at the torso (and 2 degrees of freedom at the neck). Of these, only the 7 DOF of a single arm has been used while the torso joint was fixed. The mobile base has a holonomic configuration, provided by 4 mecanuum wheels. Thus, the considered kinematic chain had a total of 10 degrees of freedom. A Robotiq FT-300s force-torque sensor has been mounted on the wrist of the robot, and a Robotiq 2F-85 gripper is placed on top of it. The system runs Ubuntu 22.04 with a PREEMPT_RT patched kernel and runs ROS 2 Humble framework [79]. The elastoplastic controller has been implemented in C++ as the `ros2_control` plugin, utilizing the `eiquadprog` library [19] to solve the optimization problem. The source code is available at [36].

To showcase the use of the algorithm in transport scenarios and assess its performance, the following experiment has been devised. A rigid cardboard plate is tightly grasped by the

robot on one side and by the human operator on the other. The robot is commanded to follow a linear trajectory along the X-axis of the world frame. From the operator's perspective, the trajectory is divided into two parts: in the first half of the task, the operator is required to simply follow the robot, as a follower agent. This phase is designed to highlight the trajectory tracking capabilities of the controller as well as the compliance provided by the elastic mode. In the second half, the operator will apply a force to the payload in order to pull the robot and have it perform an obstacle avoidance maneuver. During this phase, the control algorithm is expected to transition into plastic behavior, allowing the operator to become the leading agent of the formation. Figure 3.5 gives a graphical representation of the design of the experiment.

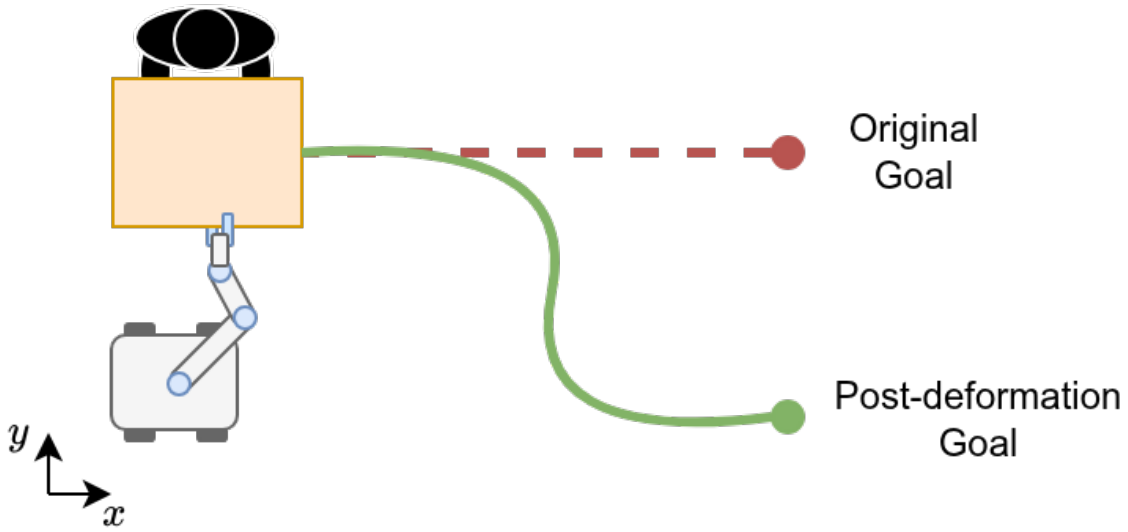


Figure 3.5 Graphical representation of the experiment

3.6.2 Experimental Setup Limits

Due to the lightweight structure of the manipulator and the high compliance at the joint level provided by the Series Elastic Actuators, which approximate the joints as a series of integrators and resonances, the bandwidth of the admittance controller was kept low, particularly by maintaining limited values for the stiffness. More details on the parameters and the tuning are presented in 3.7.5. Restrictions on the acceleration of the mobile base were also implemented to limit the frequencies excited by its movements and avoid resonances. This restriction also has the benefit of highlighting the reorganization of joint priorities when reaching kinematic limits, as presented in the following sections.

3.6.3 Assessment and Benchmarking

During the experimental study, in order to maintain simplicity and ease of operation, no additional indices were introduced. Nonetheless, the following aspects were investigated:

- the tracking accuracy of the Cartesian reference trajectory
- the response during elastic and plastic behaviors
- the trajectory recovery after the introduction of a plastic offset

To validate the behavior, the control algorithm is compared with the controller presented in [119]. This is a multi-modal controller for mobile manipulators that can exhibit multiple behaviors. Each behavior is formulated as a strict HQP with minimization variables: the joint acceleration, the joint desired torques, and the contact force at the end-effector. These variables are constrained by the robot's dynamic model. Among the various modes in this control architecture, during the experiments, only those modalities that have counterparts in the elastoplastic algorithm have been considered, namely motion tracking and admittance. Hereafter, the mathematical formulation of the optimization problems associated with each mode is briefly summarized. Defining $\chi = [\ddot{q},\ \tau,\ f_c]^T$, both modalities are expressed through the following HQP:

$$\begin{aligned} \text{lex}\min_{\chi} \quad & \{T_1(\chi), T_2(\chi), T_3(\chi)\} \\ s.t. \quad & \begin{bmatrix} M & \mathbb{I} & J^{-1} \end{bmatrix} \chi = -C - g \\ & A\chi = b \\ & \chi \in [\chi_{\min}, \chi_{\max}] \\ & q \in Q_{limits}, \dot{q} \in \dot{Q}_{limits} \end{aligned} \tag{3.26}$$

where M is the inertia matrix, $C = c\dot{q}$ is the Coriolis term and g is the gravitational term. The motion tracking mode consider the following Stack-of-Task

$$T_1(\chi) = \|J\ddot{q} + \dot{J}\dot{q} - \ddot{x}_r + K_p(x - x_r) + K_v(\dot{x} - \dot{x}_r) - M^{-1} f_c\|^2 \tag{3.27}$$

$$T_2(\chi) = \|\tau - g\|^2 \tag{3.28}$$

$$T_3(\chi) = \|\dot{q}\|^2 \tag{3.29}$$

$$A = [0 \quad 0 \quad \mathbb{I}] \qquad b = f_{meas} \tag{3.30}$$

where (3.27) consider an impedance trajectory tracking task, similar to the one defined in (3.18), (3.28) is the gravity compensation task and (3.29) is the regularization task. When

under the effect of external forces, the admittance task replace the trajectory tracking Stack-of-Tasks with a new stack composed of

$$T_1(\chi) = \|f_c - f_{meas}\|^2 \tag{3.31}$$

alongside Eq. (3.28) and (3.29). Task (3.31) defines a soft requirement that penalizes deviations between the optimized contact force and the force measured by the force-torque sensor while allowing relaxation of force consistency to maintain feasibility of the HQP. By following Algorithm 1 of [119], the transition between the two SoT is provided based on the measured force f_{meas} at the end-effector and the desired force f_d: if $f_{meas} = f_c$, the motion tracking SoT is chosen while if $f_{meas} \neq f_c$, the admittance mode is activated.

This algorithm was chosen as a benchmark for several reasons. In fact, the benchmark algorithm:

- is specifically designed for mobile manipulators in physical human-robot interaction scenarios
- implements a whole-body controller through Hierarchical QP
- can render elastic, through the motion tracking mode, and non-elastic behaviors, by using the admittance mode

3.7 Experiments and Results

3.7.1 Assessment of Trajectory Tracking and Plastic Deformation

Figure 3.6 shows snapshots taken during the experiments, while Table 3.2 shows the tuning parameters used during the experiment. While the full HQP problem is used, the hierarchy change due to trajectory recovery, as described in Section 3.5, has been turned off during this experiment.

During the first part of the trajectory, the robot successfully follows the trajectory, with minimal deformation, as shown in Figure 3.7, which displays the Cartesian position of the end effector with respect to the world frame. At 5.5 seconds, the end effector starts to move away from the trajectory, as a result of the force exercised by the user. This shift is particularly noticeable along the Y-axis. It is important to note that even along the X-axis, a slight offset from the trajectory accumulates due to a small component of the force applied being along

Figure 3.6 Snapshots of the experiment

this axis. In fact, due to this offset, the target on this axis cannot be reached anymore, unless the trajectory recovery strategy is applied, as shown in a subsequent experiment.

By looking at Figure 3.9, which presents the evolution of the z state variable's norm, Figure 3.8, which shows the force input to the controller, and 3.10, which illustrates the velocity at joint levels, a more in-depth description of the controller behavior can be observed. First, during the initial part of the trajectory, a slight deformation of approximately 2.2 cm was generated by the non-perfect synchronization between the operator and the robot, resulting in minor forces. The effect of these deformations is applied only to the arm joints, while the base continues to track the trajectory. During the latter part, the force applied by the operator along the Y axis is sufficient to saturate $\|z\|$ and transition the controller into a plastic state. The effect of the behavioral change is also evident at the joint level, where the effect of the admittance is applied at the base. This results in a transverse velocity, comparable in magnitude to the reference one, that moves the mobile base away from the reference trajectory. Due to the presence of vertical forces and the impossibility of the base moving upwards or downwards, the QP problem assigns this component of the response again to the arm joint. Another situation in which this change in priorities among the joints can occur is due to the limitations of the mobile base. In fact, mobile robots usually have far lower bounds on velocities and accelerations compared to manipulators. If these limits are

reached during a task, the QP program attempts to compensate for this limitation by adjusting the movement of available joints, specifically those of the arm.

Table 3.2
EXPERIMENTS PARAMETERS

Parameters	**Value**	**Unit**
m_x, m_y, m_z	1	kg
σ_0	30	N/m
σ_1	0	Ns/m
σ_2	30	Ns/m
z_{ba}	0.07	m
z_{ss}	0.1	m
w_h	2	s
R_v	1	J

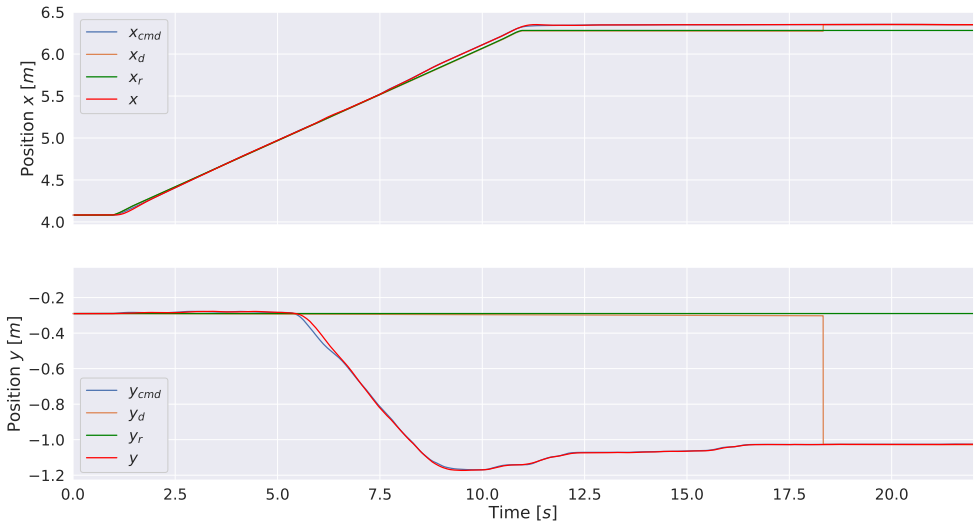


Figure 3.7 Trajectory tracking and plastic deformation: Cartesian position of the end effector. Reference trajectory in green, computed Cartesian reference in orange, forward kinematics obtained from the control action values in blue, and forward kinematics of the actual robot position in red.

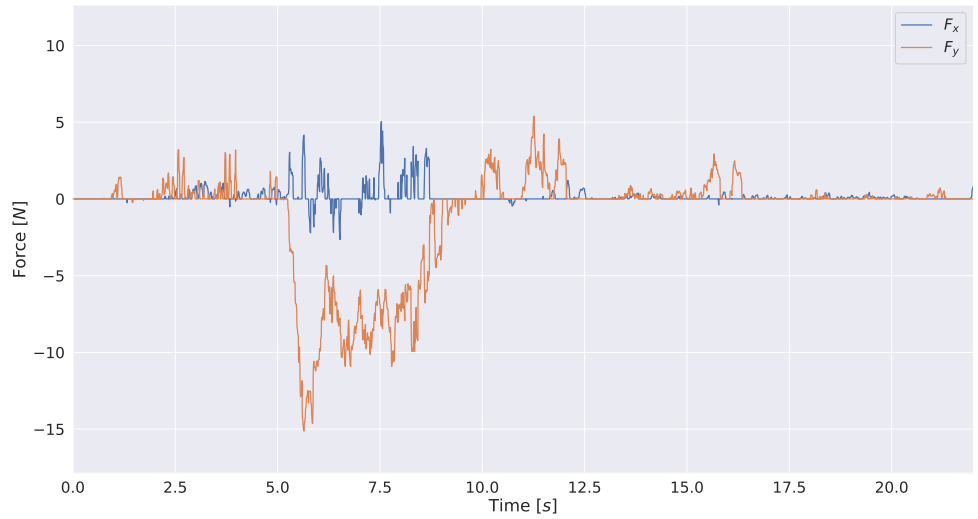


Figure 3.8 Trajectory tracking and plastic deformation: Filtered wrench provided to the elastoplastic controller.

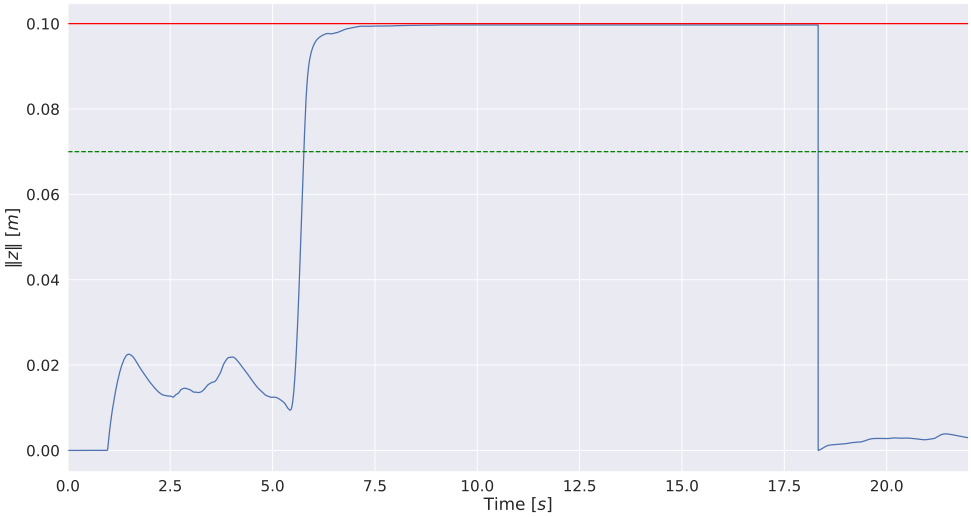


Figure 3.9 Trajectory tracking and plastic deformation: norm of state variable vector **z**. Values of z_{ba} and z_{ss} are represented as dashed green line and solid red line, respectively.

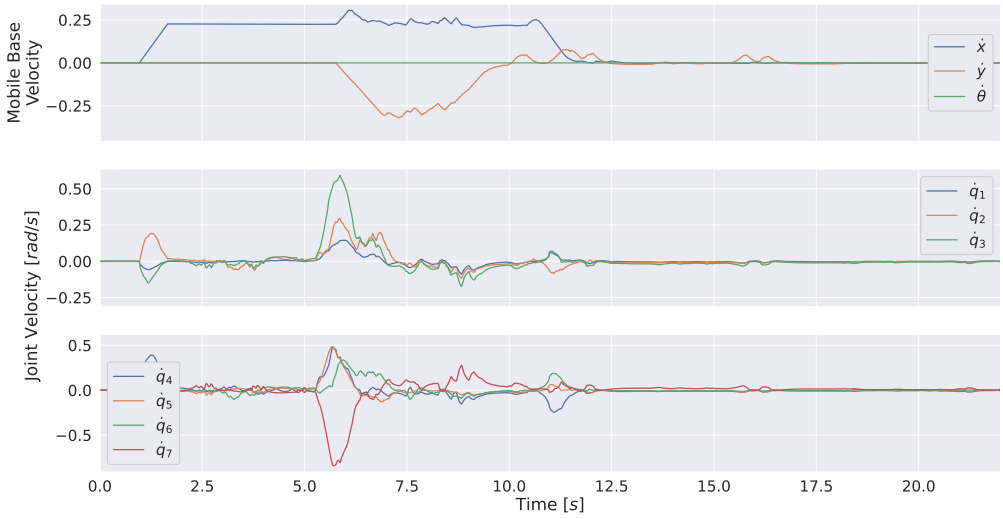


Figure 3.10 Trajectory tracking and plastic deformation: Joint velocities. The top plot represents the velocities of the mobile base expressed in the world frame, while the remaining two plots represent the joint velocities of the manipulator.

3.7.2 Assessment of Trajectory Tracking and Recovery

To show the behavior of the recovery strategy presented in 3.5, a second experiment is performed. Following the design in 3.6.1, the rigid cardboard is rigidly grasped by the robot and a human operator and transported. The robot is tasked to follow a predefined trajectory. During the first part of the trajectory, the operator will follow the robot, while during the second part, they will apply a force to the payload to pull the robot away from the path. As shown in Figure 3.11, the behavior during tracking phase and deformation phases are similar to the one exhibited in the previous experiment: The robot follows the predefined Cartesian trajectory up until 6 seconds into the experiment where the operator exert a force, as shown in Figure 3.12, that apply a plastic deformation to the Cartesian trajectory. No meaningful force is applied after time t =11 s and, at t = 13 s, the reset condition becomes true, and the controller state is reverted to the elastic one. The HQP that computes the joint acceleration is then updated as described in 3.5, with task (3.25) raised to the higher level of the hierarchy. The evolution of the desired Cartesian reference x_d, introduced in 3.5, can be seen in Figure 3.11. During the initial trajectory tracking segment, as well as during the plastic deformation of the trajectory, $\dot{x}_d = \dot{x}_r$ thus forwarding the reference trajectory to the following layers of the hierarchy, allowing the admittance layer to follow the original trajectory and, if necessary, to introduce temporary or non-temporary offset with respect to x_r. At t =13 s, with the promotion of task (3.25), the reference x_d guides the Cartesian position of the end effector on the old trajectory to cancel the plastic offset, providing the tracking layer with a suitable reference. It should be noted that, since the controller is in its elastic state, the elastic action is computed with respect to x_d and not the original reference x_r.

3.7.3 Benchmarking Algorithm Experiments

Figure 3.15 and 3.16 show, respectively, the Cartesian position and the wrench applied to the benchmark controller. From the trajectory starting at time t =2.2 s to t =8.7 s, the Cartesian trajectory provided to the robot is tracked without human intervention. At t =8.7 s, a force is detected by the end effector, which leads to the transition of the controller in what [119] called “admittance-like” behavior. For the following 3.5 s, the admittance mode objective is to compensate for the force applied, resulting in a response similar to the plastic behavior of the elastoplastic controller. When f_{meas} returns to zero, the controller switches back to motion tracking mode. Due to the presence of an elastic term in the motion tracking task (3.27), the end-effector is directly pulled back onto the trajectory, following a standard second-order response.

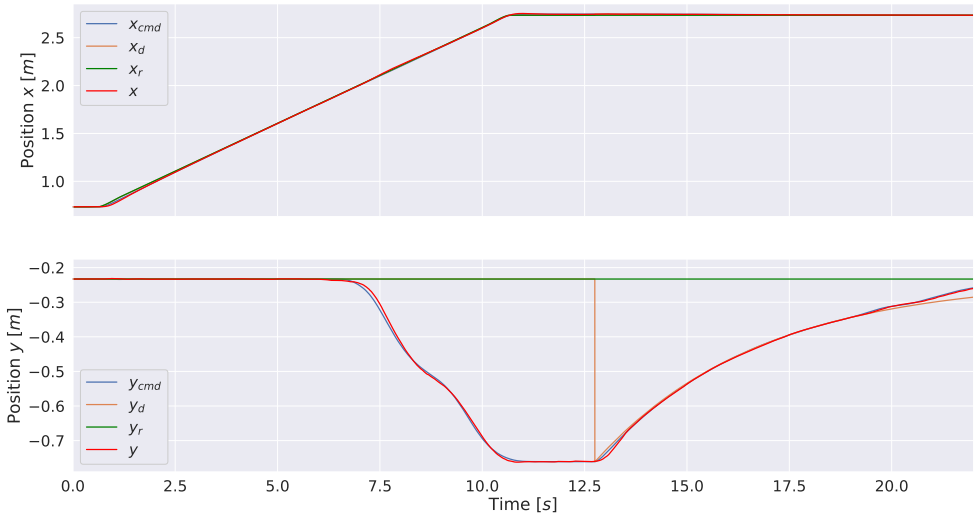


Figure 3.11 Trajectory recovery: Cartesian position of the end effector. Reference trajectory in green, computed Cartesian reference in orange, forward kinematics obtained from the control action values in blue, and forward kinematics of the actual robot position in red

3.7.4 Comparative Analysis

By focusing on the trajectory tracking capabilities, the controllers provide comparable capabilities, as they implement equivalent formulation of the same tracking task. In fact, in the elastoplastic controller, for $\alpha(z) = 0$, the state equation (3.13) simplifies to $\dot{z} = \dot{x}$, resulting in a spring-damper admittance behavior on the tracking.

The application of a significant force on the end-effector, while having in both cases the effect of switching into a "position independent" response behavior. In [119], the admittance mode depends only on the force measured, resulting in a response completely independent from the trajectory. This effect can be seen in the top plot of Figure 3.15, where the movement along the X axis, required by the reference trajectory, is interrupted for the time during which the Y-axis force is applied on the end-effector. So the trajectory tracking behavior and the admittance behavior are, in fact, completely separated from a high-level task perspective, and the switch to the admittance mode represent an abrupt interruption of the tracking task instead of a trajectory correction. By contrast, the trajectory tracking property of the elastoplastic controller is maintained during the plastic movement. This is observable in Figure 3.11, where the Y-axis force produces a deviation from the original trajectory only along such axis while the main trajectory direction is unaffected.

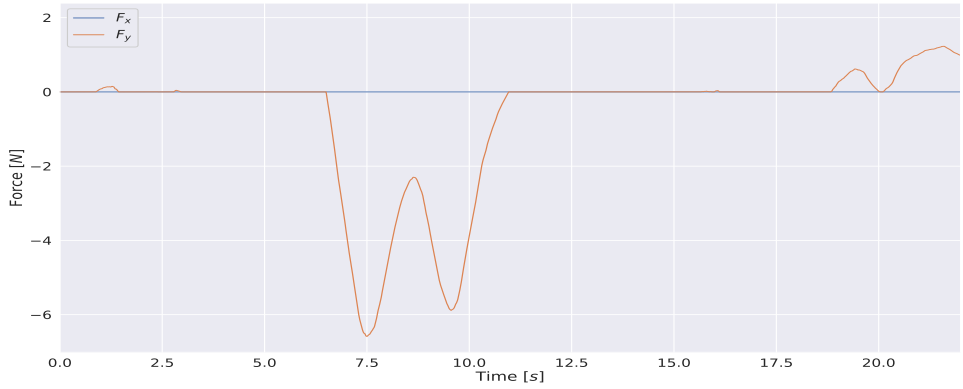


Figure 3.12 Trajectory recover: Filtered wrench provided to the elastoplastic controller

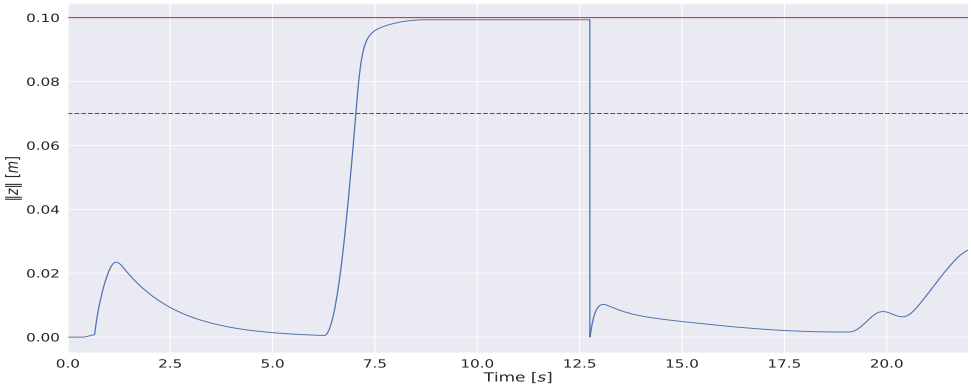


Figure 3.13 Trajectory recover: norm of state variable vector **z**. Values of z_{ba} and z_{ss} are represented as dashed green line and solid red line, respectively.

A key qualitative difference with respect to the benchmark controller in [119] concerns how trajectory re-entry is managed after interaction. In [119], Algorithm 1 switches mode to trajectory tracking when no force is detected at the end effector. This re-establishes an elastic dependence on the position error with respect to the trajectory, effectively pulling the robot back through a fixed spring–damper response whenever the mode switch occurs. However, the absence of an explicit buffer time between force disappearance and mode switching can lead to repeated commutations around the switching condition, especially when the contact wrench is near the detection threshold.

In contrast, the approach proposed in this thesis performs the trajectory re-entry at reference level (via (3.18)), which decouples compliant task execution from the evolution of the desired reference x_d. This separation enables recovery behaviors that remain compliant

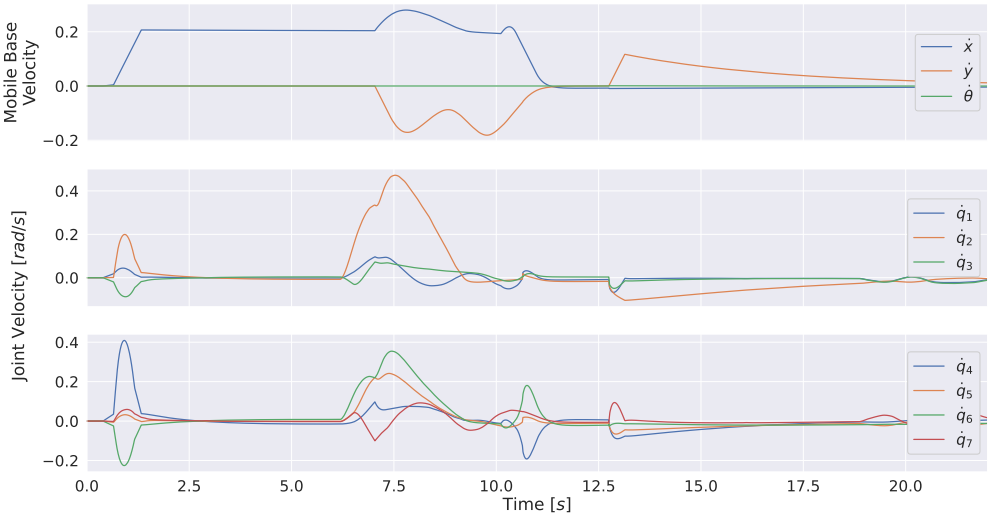


Figure 3.14 Trajectory recovery: Joint velocities. The top plot shows the velocity of the mobile base expressed in the world frame, while the remaining two plots represent the joint velocities of the manipulator.

around the converging reference rather than imposing a fixed comeback behavior in task space. This effect is particularly visible in the latter portion of the recovery experiment, after the recovery activation, where non-negligible forces can still be applied while the controller renders a compliant behavior around the converging desired reference. Moreover, recovery is only enabled once the reset condition 2.4.1 is satisfied. In practice, the windowed evaluation prevents multiple switches within short time intervals and mitigates undesirable oscillations between interaction and tracking behaviors.

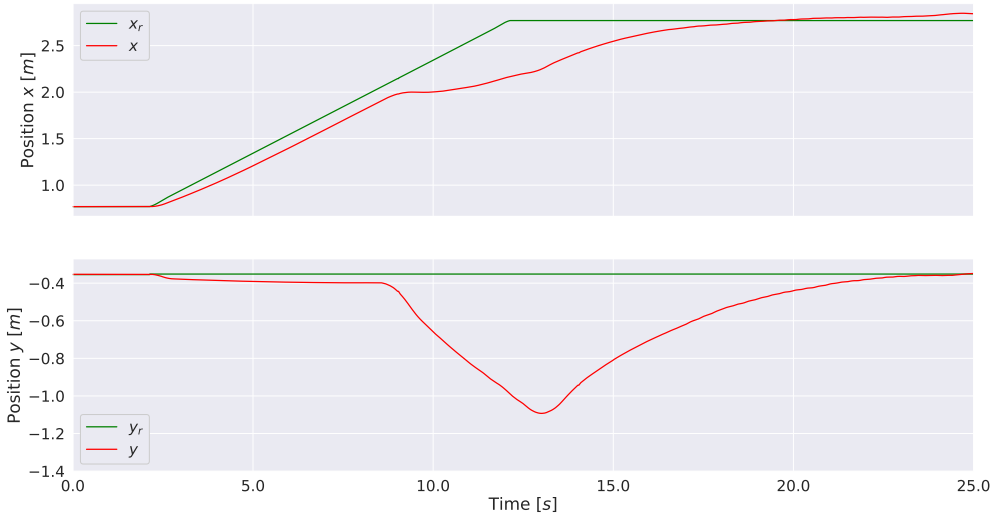


Figure 3.15 Benchmark Controller: Cartesian position of the end effector. Reference trajectory in green and forward kinematics of the actual robot position in red.

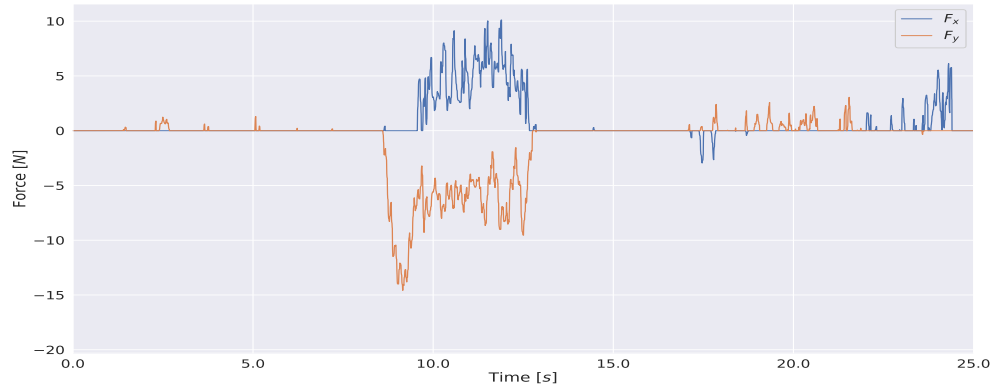


Figure 3.16 Benchmark Controller: Filtered wrench provided to the benchmark controller.

3.7.5 Discussion

To obtain a coherent and suitable response for physical human-robot interaction, several factors must be taken into account when using the elastoplastic controller. At first, the controller requires the tuning of multiple parameters. A series of tests was conducted to obtain a set of parameters (Table 3.2) that was user-pleasing while also being suitable for the task and compatible with the hardware. Table 3.3 summarizes the qualitative influence of each parameter.

Among them, σ_0 is a critical quantity: it directly sets the elastic stiffness perceived by the operator and indirectly constrains several other design choices. As described in [30], σ_0, the Coulomb breakaway force f_{ba} and the breakaway displacement are related by the relation $f_{ba}/\sigma_0 = z_{ba}$. This relation provides a convenient interpretation for tuning, as f_{ba} can be treated as the yield force threshold. In contrast, z_{ba} represents the minimum deformation required before plastic deformation occurs.

In addition, for purely elastic behavior, the overall damping σ_d of the model is $\sigma_d = \sigma_1 + \sigma_2$. To have an elastic damped response, the following should be satisfied

$$\sigma_d \geq \sigma_{d,c} = 2\sqrt{M\sigma_0}$$

where $\sigma_{d,c}$ is the critical damping.

Figure 3.17 illustrates the influence of σ_0 on the evolution of state variable z under step inputs. For fixed z_{ba} and $z_{ss} = z_{ba}/\gamma$, $\gamma < 1$, and with σ_1 and σ_2 selected to obtain critically damped behavior for unit mass, the same force magnitude can lead to opposite behaviors (remaining elastic opposed to transitioning to plastic) depending on the chosen stiffness. This emphasizes that stiffness selection cannot be decoupled from the intended interaction task and should be tuned jointly with the desired yield threshold and the expected magnitude of human-applied forces.

A second critical tuning aspect is the ratio $\gamma = z_{ba}/z_{ss}$ which governs the slope of the sigmoid $\alpha(z)$ and therefore the rate at which stiffness is reduced when transitioning from elastic to plastic behavior. For a fixed input force, γ values closer to one produce a faster stiffness collapse, yielding more immediate plasticity. Conversely, smaller γ values spread the transition over a larger deformation range, providing smoother but slower commutation.

In the limit case in which $z_{ba} = z_{ss}$, the sigmoid degenerates into a step thus nullifying the elastic effect in a single timestep and resulting into a sudden drop in resistant force by the robot, due to the lack of time given to the remaining components of the admittance system (damping and inertia) to compensate the disappeared stiffness. At the opposite extreme,

selecting $z_{ba} = 0$ results in preventing stiction, and thus elastic behavior, from being rendered. Therefore, γ should be tuned as an explicit trade-off between transition smoothness and responsiveness, based on the intended use case.

Another consideration concerns the mobile base state used by the whole-body admittance layer. In practice, estimating the base pose by open-loop integration of commanded (or even measured) accelerations/velocities is not reliable over extended horizons. Unmodeled dynamics, parametric uncertainty (e.g., wheel radius and wheelbase errors), actuation limits, wheel slip and skidding, and external disturbances introduce non-negligible inconsistencies between the commanded motion and the realized motion. These discrepancies accumulate during integration, resulting in drift in the internally propagated pose that ultimately compromises the consistency of task-space error evaluation and constraint enforcement. This limitation is further exacerbated on mobile manipulators due to the presence of a moving arm mounted on the platform. Arm motion modifies the composite inertial properties of the system and can shift the center of mass, thereby changing normal forces at the wheel–ground interface. The resulting variations in contact conditions affect friction margins and increase the likelihood of slip during acceleration, braking, and turning maneuvers. Consequently, the introduction of localization techniques based on internal sensing (e.g., sensor fusion of multi-source odometry measurement using Kalman Filter-derived algorithms) and, especially for long-term operations, external sensing is required to maintain an adequate level of accuracy throughout the task.

Table 3.3
PARAMETERS TUNING CONSIDERATIONS

Parameter	
M	Regulates the inertial response of the controller. As in standard admittance control, larger inertia results in slower motions. Alongside σ_2, the plastic behavior remains unchanged, defining its response.
σ_2	Regulates the absolute viscous damping, thus modulating the velocity of the controller as a response to external forces. In elastic mode, it is part of the damping term $D = \sigma_1 + \sigma_2$, while in plastic mode, it defines the overall damping behavior. It is not involved in the impedance variation.
σ_1	represents the damping of the LuGre bristle. In elastic mode, it is part of the damping term, while it rapidly becomes negligible during the commutation to plastic as $\dot{z} \to 0$.
σ_0	Elastic coefficient. Define the stiffness of the controller. Related to z_{ba} by the relationship $f_{ba}/\sigma_0 = z_{ba}$, with f_{ba} the force threshold above which the controller starts producing plastic offsets.
z_{ba}	Minimum deformation of the reference path required to start the elastic to plastic transition. Related to stiffness parameters by the relationship $f_{ba}/\sigma_0 = z_{ba}$, with f_{ba} the force threshold above which the controller starts producing plastic offsets.
z_{ss}	Maximum value reachable by the state variable z. At equal input forces, higher values of z_{ss}/z_{ba} require a longer time to complete the switch from elastic to plastic behavior, while providing smoother transitions.
w_h	Window size to compute the reset work index. For low values of R_v, define the minimum time required for the controller to be reset when the robot is stationary ($\dot{x} = 0$) or no force is applied ($F_{in} = 0$).
R_v	Work reset index threshold. Low values assure that the controller resets only if it is stationary ($\dot{x} = 0$) or it is not interacting with the environment ($F_{in} = 0$).

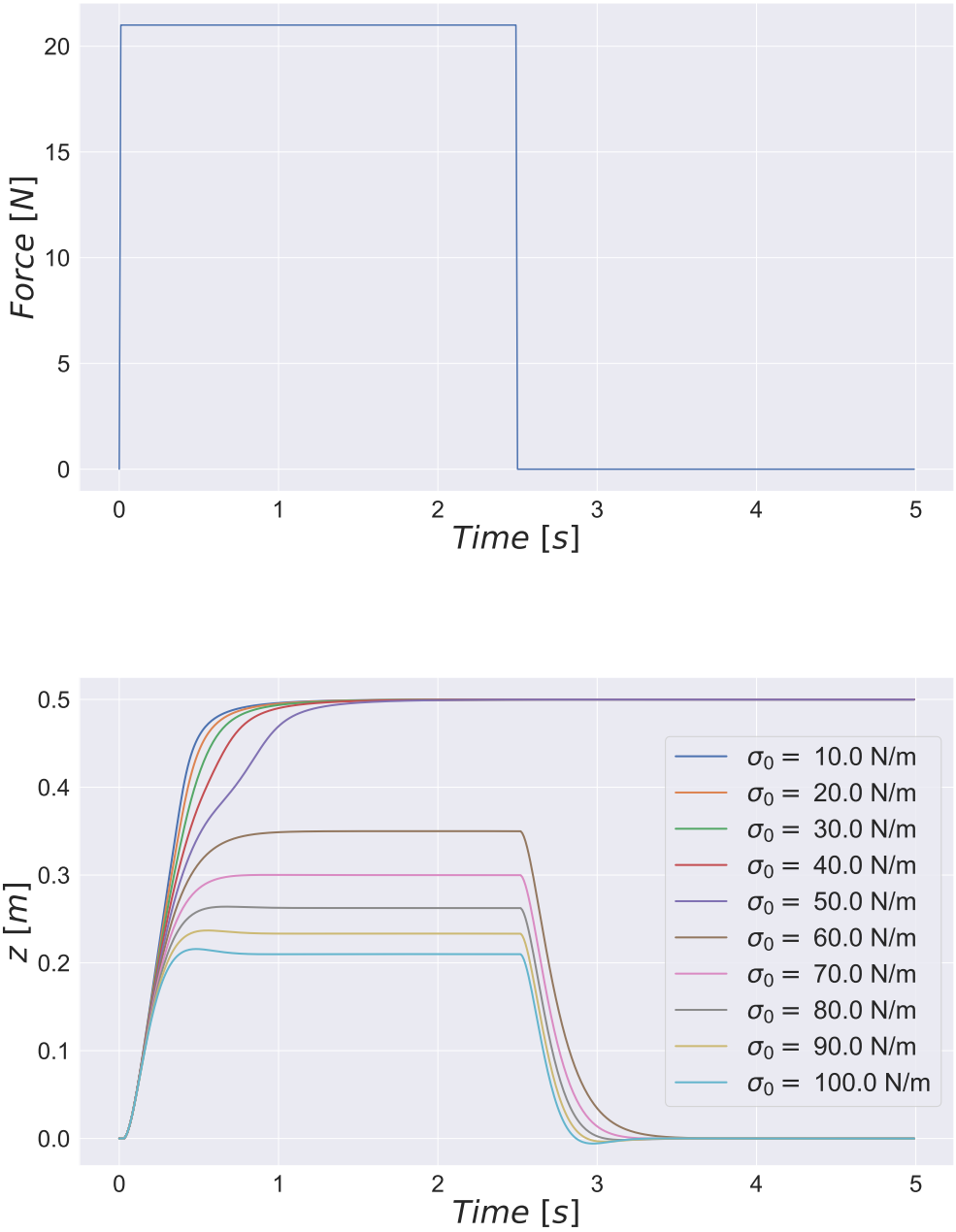


Figure 3.17 Effect of different values of σ_0 on the dynamics of z

3.8 Summary

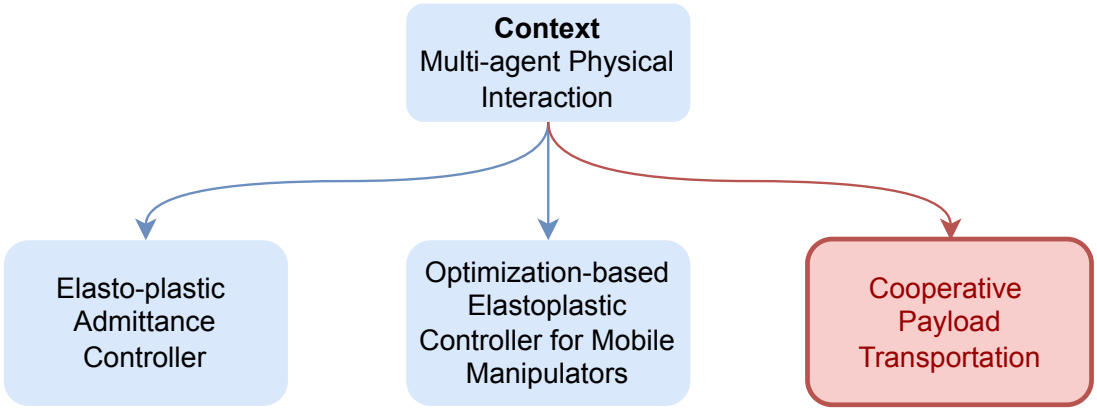
This work, presented in this Chapter, extends the elasto-plastic interaction control paradigm from fixed-base manipulators to whole-body mobile manipulation, targeting collaborative and contact-rich tasks where both trajectory fidelity and operator-driven adaptation are required. The proposed approach integrates an elasto-plastic admittance behavior within a whole-body control architecture formulated as a hierarchical quadratic program, thereby enabling consistent distribution of motion and interaction across the mobile base and the arm while maintaining feasibility under hardware limits and kinematic constraints.

A key element of the chapter is the introduction of an optimization-centric controller design that preserves continuous trajectory execution while allowing permanent path offsets under sufficiently high interaction forces. This behavior is paired with an arbitration policy that reshapes redundancy usage as the interaction regime changes: arm motion is preferred when small, reversible deformations are sufficient, whereas the mobile base is promoted for larger, sustained deviations that benefit from extended workspace and coordinated whole-body motion. In addition, a trajectory recovery strategy is presented to reintroduce convergence toward the nominal path after a permanent offset has been created by optimizing an intermediate Cartesian reference within the hierarchical program rather than enforcing an abrupt return through the interaction law itself.

The experimental campaign on a mobile manipulator in a collaborative payload transport scenario validated the controller's ability to track a nominal path with minimal deformation during leader-like operation, transition to a follower-guided behavior when the human induces significant deviation, and recover the nominal trajectory after the interaction subsides. A comparative evaluation against a multi-modal whole-body controller highlighted the practical advantage of maintaining trajectory progress during interaction and achieving smoother, more controllable re-entry through reference-level recovery.

Chapter 4

Control Shared Architecture for Cooperative Transport



This chapter presents a comprehensive leader–follower control framework designed to support cooperative transport tasks performed by teams of mobile manipulators. The proposed methodology integrates a centralized cooperative planner with a decentralized follower control scheme, offering a unified solution to the inherent challenges of multi-robot coordination in dynamic and potentially unstructured environments. At the core of the approach is a mapping strategy in which each robot contributes a local 2D costmap; these individual representations are fused into a coherent 3D environmental model that supports global path planning. Standard graph-based algorithms, such as A^* or Θ^*, can then be applied to compute collision-free and globally consistent trajectories for the leader agent. Once a path is generated, follower robots use a decentralized control strategy to maintain the required formation geometry and payload constraints, ensuring that the shared object is transported safely and smoothly. The full framework is validated in a real-world experimental setup

involving two mobile manipulators transporting a rigid object, demonstrating its feasibility, robustness, and practical applicability.

The Chapter is structured as follows:

- **Section 4.1** introduces the cooperative transport problem for teams of mobile manipulators and motivates a leader–follower strategy that combines centralized planning with decentralized execution. It summarizes the intended operating conditions, key design goals (real-time performance and robustness in dynamic environments), and the main contributions, including integration with a standard ROS2 navigation stack.

- **Section 4.2**: establishes the foundational concepts required by the framework. It reviews map representations used in robotics navigation (occupancy grids and costmaps), and frames path planning as a costmap-based problem compatible with common graph-search planners. It then positions formation control within the broader multi-robot literature, discusses the main paradigms (behavior-based, virtual structure, and leader-follower), and motivates the leader-follower choice in terms of scalability, communication requirements, and practical deployment trade-offs.

- **Section 4.3**: presents the augmented costmap construction used to make single-robot path planning feasible for an entire formation. It explains how local costmaps from each robot are transformed into a common reference, fused into a formation-level representation, and augmented to account for formation geometry. The section also describes the multi-layer (orientation-discretized) representation and how standard planners can operate on it, including the post-processing step that converts a multi-layer solution into a usable planar trajectory, as well as the rationale for periodic updates in changing environments.

- **Section 4.4**: details the leader-follower formation control architecture that executes the planned motion. It specifies the role of the leader in tracking the global trajectory and broadcasting the information needed by followers, and describes the follower control structure based on relative-pose regulation.

- **Section 4.5**: introduces the compliance strategy for cooperative manipulation used to mitigate internal loading caused by residual errors.

- **Section 4.6**: reports experimental validation and practical considerations. Simulation results cover both synthetic-map benchmarking and physics-based evaluation, with emphasis on computational performance of costmap fusion and planning, and on

formation behavior during navigation. Real-world experiments describe the dual-robot hardware setup, the localization and path-tracking components used on each platform, and the cooperative transport demonstration with a shared payload. The closing discussion highlights critical deployment factors such as localization stability, costmap quality and self-detection issues in mapping.

- **Section 4.7** concludes the Chapter and summarize the methodology and results presented.

The content of this Chapter is based on the work in [89].

4.1 Introduction

4.1.1 Context

Recent advancements in robotics, accelerated by the technological shifts associated with Industry 4.0 and the emerging concepts of Industry 5.0, have significantly broadened the deployment of autonomous systems across a wide range of sectors. Mobile robots are now routinely employed in warehouses, large-scale logistics, retail operations, healthcare environments, and service industries, where they perform tasks such as material handling, item delivery, and collaborative assistance. However, despite their widespread adoption and growing capabilities, conventional mobile platforms often remain limited in key aspects: payload capacity tends to be relatively small, the ability to manipulate objects autonomously is constrained or absent, and task execution frequently requires external mechanical mechanisms or human intervention.

The integration of mobile manipulators, robotic systems that combine a mobile base with one or more articulated arms, offers a promising strategy to overcome these limitations. Such robots can navigate complex spaces, interact directly with objects, and execute sophisticated manipulation tasks. Yet, when multiple mobile manipulators must collaborate, particularly in cooperative transport scenarios, new challenges arise. Transporting a rigid object requires all agents to maintain strict geometric constraints: inter-robot distances, orientations, and motion profiles must remain synchronized to prevent excessive forces on the payload or destabilization of the formation. These requirements go beyond the capabilities of traditional formation control techniques, which typically assume homogeneous fleets with no shared physical constraints. Additional difficulties are introduced by real-world factors such as

Figure 4.1 Typical example of an industrial mobile manipulator with a lightweight collaborative robot mounted on a mobile platform. (Credits to Neobotix)

imperfect localization, uncertain perception, communication delays, and dynamic obstacles, all of which can degrade performance and compromise safety if not properly managed.

In this context, advancing cooperative control strategies that are both robust and computationally efficient is essential. The framework introduced in this chapter addresses these challenges by combining multi-robot mapping, centralized planning, and decentralized control into a cohesive architecture tailored for cooperative manipulation tasks.

4.1.2 Contributions and Assumptions

This work makes several contributions to the field of cooperative mobile manipulation:

1. **Costmap-based Multi-Robot Formation Representation**: A costmap-based framework is presented, that combines information from multiple robots to generate a unified environmental map, ensuring collision-free path planning for all agents in the formation.

2. **Decentralized Control Scheme**: A distributed control strategy is introduced for computing velocity commands for follower agents in a leader-follower configuration, relying on real-time localization data and leader commands.

3. **Planner and Control Agnosticism**: The proposed method is agnostic to both the underlying path-planning algorithm and the leader's velocity controller, allowing compatibility with state-of-the-art graph-based planners and diverse control schemes.

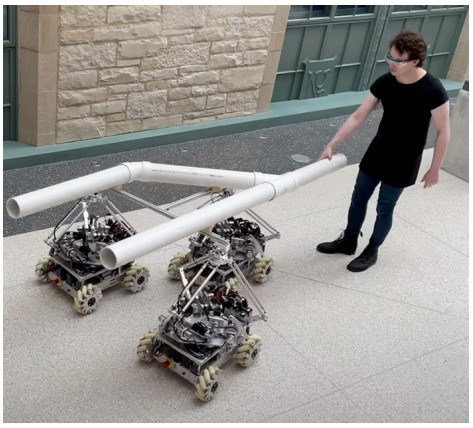


Figure 4.2 Team-based robotics system could support humans working in construction zones, manufacturing plants. (Credits to: Center for Robotics and Biosystems, Northwestern University)

4. **Open Source Implementation**: An open-source implementation of the framework, compatible with the popular Nav2 navigation stack, is provided to facilitate adoption and further research.

These contributions provide a practical and scalable approach to cooperative transport in multi-robot systems, with a particular emphasis on real-time performance and adaptability in dynamic environments.

4.2 Fundamentals

4.2.1 Costmap-based Path Planning

While industrial manipulators are mounted on fixed bases and operate in limited, and usually well-structured, workspaces, mobile robots are designed to work across large areas that may vary significantly in terms of complexity and prior knowledge. These environments can be fully known a priori, as in industrial or warehouse contexts where the layout is fixed and predictable, or completely unknown, as in exploration and search-and-rescue missions. In both cases, autonomous navigation requires the robot to estimate its own position, detect and localize relevant targets or obstacles, and compute feasible and safe paths. To achieve this, navigation algorithms frequently rely on the concept of maps, which provide a structured representation of the surrounding environment [121]. Formally, a map can be defined as a collection:

$$m = m_1, \ldots, m_i, \ldots, m_N \tag{4.1}$$

of N properties of the environment. If the element m_i corresponds to a distinct feature of the environment, storing both its physical properties and its Cartesian coordinates, the resulting representation is referred to as a feature-based map. Such maps are commonly used in SLAM systems that rely on identifiable landmarks such as corners, edges, or fiducial markers [28, 9]. Conversely, if each element m_i represents a specific location in space and an associated set of properties, the map is termed a location-based map. This second category includes grid-based methods that discretize the environment into cells and annotate each cell with relevant information.

A widely used form of location-based representation is the occupancy grid, originally introduced in [32, 86]. In an occupancy grid, the environment is represented as a 2D lattice where each cell is assigned a binary or probabilistic state indicating whether it is free or occupied. Occupancy grids have proven highly effective due to their simplicity, scalability, and compatibility with a broad range of sensing modalities. However, their expressiveness can be significantly expanded beyond a binary classification. By allowing each cell to encode richer information as costs, such as proximity to obstacles, hazardous areas, or preferred corridors, the representation becomes a costmap. The concept of costmaps is widely employed in modern robotic navigation systems, including ROS-based frameworks such as the Navigation Stack and Nav2 [82, 76]. In a costmap, each cell is associated with a value $c \in [c_{min}, c_{max}]$, where c_{min} denotes a completely free cell, c_{max} indicates an occupied or forbidden region, and intermediate values represent soft constraints guiding robot motion. These soft constraints can be derived from multiple sources including raw sensor readings (e.g., LiDAR, depth cameras, ultrasonic sensors), safety margins around humans or sensitive equipment [63], or manually defined restricted zones specified by operators. Through this enriched representation, costmaps provide a flexible and powerful tool for encoding navigation-relevant information.

In modern navigation frameworks, costmaps are typically not updated monolithically. Instead, a layer-based structure is employed to ensure modularity, scalability, and clearer semantic interpretation. This architecture was formalized and popularized through the layered costmap design in ROS navigation [71]. Each layer corresponds to an independent costmap generated from a single source and representing a unique environmental concept—for example, a static layer describing immovable obstacles, an obstacle layer updated from LiDAR scans, or a social layer encoding proxemic constraints around humans [61]. These layers are stacked hierarchically, and the final costmap is obtained by combining them sequentially according to a predefined update order. This hierarchical approach ensures deterministic and reproducible updates to the global costmap and enables certain layers to

depend explicitly on the outputs of lower layers, such as inflation layers that expand obstacle boundaries based on the underlying occupancy information.

The grid structure of occupancy grids and costmaps lends itself naturally to the application of graph-based pathfinding algorithms for solving the path planning problem. By treating each cell as a node in a graph and each feasible transition as an edge, classical methods such as Dijkstra's algorithm, A*, or Θ^* can be applied directly [47, 91]. In this context, the expressiveness of costmaps greatly enhances navigation performance: costs can be inflated around obstacles to create smooth safety buffers [105], high-cost regions can discourage traversal through undesirable areas, and low-cost corridors can guide the robot along preferred or safer routes.

4.2.2 Formation control

As scale and complexity of tasks grows, a single robot is rendered insufficient to efficiently perform the activities. The introduction of additional robots in the environment allows to relax the requirement of the robot and carry out operations impossible for single robots but, at the same, time introduce a whole new level of challenges, related to synchronization, shared resources and traffic control. Among those, especially for mobile robots and drones, can be required to maintain a somewhat rigid formation to perform payload transport or display particular shapes. Algorithms that enable the ensure the robots are distributed following a predefined logic falls under the family of formation control algorithms. Traditionally, state-of-the-art formation control techniques are grouped in three categories: Behavior-based, leader-follower and virtual structures.

Behavior-based approaches are decentralized techniques in which every robot can behave according to a set of predefined behaviors. The interaction of these behaviors among the robots define the formation overall response. Due to the high degree of decentralization and independence of each robot w.r.t the rest of the formation, these methods are usually employed in large-scale multi-robot systems. Conversely, virtual-structures approaches are heavily centralized methods in which the formation is considered as a unique structure, where every agent in the formation is tightly connected with a number of peers. A centralized system compute the behavior of the formation as a unique body and, from that, derive the appropriate actions that each robot have to follow it. An intermediate approch is the leader-follower method. One or more agents are deemed "leader" of the formation while the rest of the formation agents are the followers. Leaders robot choose a behavior to solve the task while, at the same time, produce a reference for the follower agents. Leader behavior

usually takes into account the state of the whole formation. Followers then determine their own behavior based on the reference provided by the leader, by their own state and the prescribed geometric configuration of the formation. They do not consider other elements of the formation, if not as obstacles.

Leader–follower methods offer several practical benefits. A key advantage is that these methods allow for a clear hierarchical structure in solving coordination problems: global planning and decision-making can be centralized at the leader level, while followers simply need to execute local tracking controllers. This significantly lowers the computational complexity for the follower agents and simplifies controller design. Another benefit is that this approach easily integrates with existing navigation and motion-planning frameworks, since the leader can use standard single-robot planners, and the followers can track the resulting trajectory through relative pose regulation. Additionally, leader–follower schemes can function with limited communication requirements, particularly when followers rely on relative measurements rather than complete state broadcasts, making this method suitable for systems with constrained bandwidth.

Despite these advantages, leader–follower formation control has intrinsic limitations. One major drawback is the presence of a single point of failure: any errors in the leader's localization, perception, or planning can propagate throughout the entire formation. Communication delays or packet loss may further degrade tracking performance, potentially leading to oscillations or instability in the followers. Moreover, when followers track other followers instead of the leader directly, errors can accumulate through the leader–follower chain, reducing robustness as the formation size increases. Furthermore, leader–follower schemes provide limited flexibility for formation reconfiguration; changes in the formation geometry typically require explicit recalculation of relative offsets, which may cause temporary performance degradation.

4.3 Augmented Costmap

In multi-robot systems, particularly those operating in a leader-follower configuration, maintaining a consistent rigid formation while navigating dynamic environments is crucial. The augmented costmap method provides a mechanism for merging individual robot costmaps into a shared representation, enabling path planning that adheres to rigid formation constraints. This section presents the formulation of the augmented costmap, the merging procedure, and the path-planning methodology.

As described in 4.2.1, to navigate in the environment, each robot maintains an internal costmap that captures both static and dynamic obstacles detected by the robot itself. While in a multi-robot formation, the costmap representation is limited as provide information only on the environment seen by the single robot, thus making any plan obtained from it not suitable for every other robot in the formation. The core idea of the present planner is to obtain a costmap representation of the environment for the whole formation such that, a plan obtained from this costmap through standard single robot path planner will result in a feasible plan for the whole formation.

Consider a formation consisting of N mobile robots, with base frames denoted as M_i for $i \in [0, N-1]$, where each robot has its own internal costmap, denoted C_i. A reference frame, referred to as the leader frame L, is used as the basis for all coordinate transformations. Let ${}^{M_i}p_L \in \mathbf{R}^2$ represent the translation component of the transformation from the base frame of robot M_i to the leader frame L. The formation's costmap C is then obtained by shifting each robot's costmap C_i according to the transformation ${}^{M_i}p_L$, expressed as:

$$C = \sum_{i=0}^{N-1} {}^{M_i}p_L C_i \tag{4.2}$$

This approach effectively shifts each costmap by its corresponding translation and combines the shifted costmaps into a merged map. By positioning the leader frame L at any collision-free location within the merged costmap, a feasible path can be generated for translation-only movements. In this scenario, the collision-free path planning in the merged costmap corresponds to a valid plan for the entire formation.

However, this merged costmap is only valid for translation-only movements when the shifts between C_0 and C_i correspond to the same shifts between the robot frames M_i and L in the map frame. This indicates that rotation-based movements cannot be directly planned using the merged costmap, as the orientation of the robots is not considered while building the the costmap.

To picture this problem, consider the simple case of two robot formation, $i = \{0, 1\}$, with $M_0 = L$, i.e. the leader frame coincide with the base frame of the 0-th robot. A rotation of $\frac{\pi}{2}$ of the leader robot around its own axis result in a roto-translation of the follower M_1. Then, ${}^{M_i}p_L \neq {}^{C_0}t_{C_1}$ meaning that the costmap does not contain a meaningful representation of the obstacles viewed by the formation anymore.

To address this limitation, a new dimension is introduced into the costmap representation. Specifically, positive and negative rotations of the leader robot are discretized into steps of $\Delta\theta$, with the total rotation span represented by $K\Delta\theta$, with $K \in \mathbb{N}$. For each discrete rotation

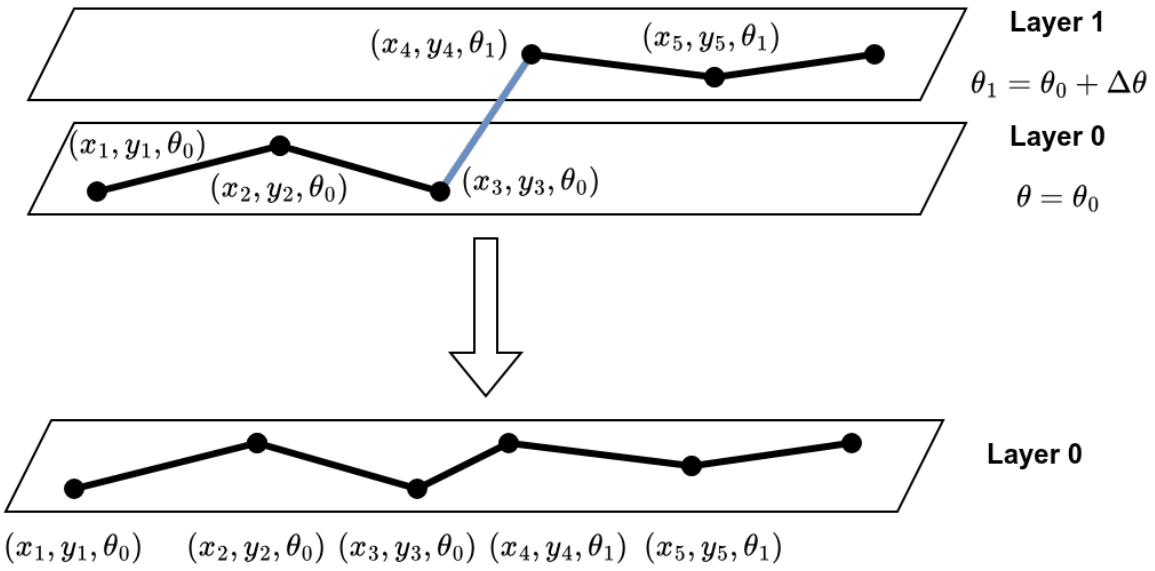


Figure 4.3 Multilayer path and squeezing procedure

angle $k\Delta\theta$, $k \in [-K, K] \subset \mathbb{N}$, Eq. (4.2) is applied, generating a new costmap. These $2K+1$ costmaps are then stacked into a 3D map, with each layer corresponding to a different rotation angle of the formation. This 3D structure represents the formation-compatible free space in the environment at various levels of rotation.

The resulting 3D map is still compatible with standard graph-based planning algorithms, such as Dijkstra's algorithm and A*. To convert this map into a weighted graph, the distance between adjacent layers is set to $q_p\Delta\theta$, where q_p is a weighting coefficient. Each cell in a given layer k is connected to corresponding cells in the adjacent layers, $k+1$ and $k-1$. Once the path is computed (potentially spanning multiple layers), a "squeezing" procedure is applied. Every intra-layer path portion is projected on layer zero while retaining the original layer orientation. This procedure reduces the 3D path to a 2D path where each section of the path is obtained from the correct augmented costmap relative to the orientation.

To ensure that the formation is responsive to dynamic changes in the environment, such as the appearance of new obstacles or changes in the formation's configuration, the merged costmap and the planned path are updated periodically, as the previous results are invalidated. This dynamic updating enables the planner to adapt to real-time changes and maintain a feasible and safe path for the entire formation.

4.4 Leader-Follower Formation Control Architecture

To apply this technique to a formation of mobile robot, the case $L = M_0$ is considered. By following this approach, the architecture is organized as follows:

1. a *leader* mobile base tracks the global reference path and generates a reference velocity command
2. each *follower* regulates its relative pose with respect to the leader through a decentralized feedback controller augmented with feed-forward action

Figure 4.4 reports the overall control architecture adopted to enforce a rigid formation among multiple mobile manipulators.

4.4.1 Leader path tracking

If $L \neq M_0$, the leader robot has to be equipped with a suitable control strategy. As presented in 4.3, the augmented costmap allows to plan a path, feasible for the whole formation, by using standard graph-based pathfinding techniques. The planning methodology guarantees that by having the leader robot strictly following this reference path, the formation will not collide with existing objects. For this reason, State of the Art controllers that can track with sufficient accuracy the reference, by keeping into account eventual non-holonomic constraints such as [77], can be used for the leader robot.

4.4.2 Relative-pose regulation for rigid formation keeping

Let ${}^{W}T_{M_0} \in SE(3)$ be the homogeneous transformation describing the pose of the leader base frame M_0 with respect to the world frame W, and let ${}^{W}T_{M_i} \in SE(3)$ be the analogous pose for the i-th follower base frame M_i. The leader pose is broadcast to all followers and used to compute the actual relative transformation between leader and follower as

$$ {}^{M_0}T_{M_i} = \left({}^{W}T_{M_0}\right)^{-1} {}^{W}T_{M_i}. \tag{4.3} $$

A desired rigid formation is encoded by a reference planar transformation ${}^{M_0}\bar{T}_{M_i}$ (constant in time for rigid formations), which specifies the nominal displacement of follower i from the leader. In the considered setting, only the planar coordinates (x, y) and the yaw Euler angle

ψ are regulated. Accordingly, a planar error vector is constructed as

$$\mathbf{e}_i = \begin{bmatrix} e_{i,x} \\ e_{i,y} \\ e_{i,\psi} \end{bmatrix}, \qquad \mathbf{e}_i = \Pi\Big(\big({}^{M_0}\bar{T}_{M_i}\big)^{-1} {}^{M_0}T_{M_i} \Big), \tag{4.4}$$

where $\Pi(\cdot)$ extracts the planar displacement and yaw mismatch from the relative transformation (e.g., by direct coordinate differencing under small-angle assumptions, or by a suitable $SE(2)$ logarithm map).

Each follower implements a decentralized PI regulator on the planar error:

$$\mathbf{u}_i^{\mathrm{fb}}(t) = K_P \mathbf{e}_i(t) + K_I \int_0^t \mathbf{e}_i(\tau)\, d\tau, \tag{4.5}$$

with K_P and K_I diagonal or full positive gain matrices. The decentralized structure implies that follower i uses only its own state and the broadcast leader information, without requiring inter-follower communication.

To improve transient response and tracking accuracy, the leader also broadcasts its velocity command $\mathbf{u}_0 \in R^3$. Since $\mathbf{u}_0$ is expressed in the leader frame, it must be transported to the follower frame consistently with rigid-body kinematics, through the adjoint representation associated with the relative transformation. Denoting by $\mathrm{Adj}(\cdot)$ the adjoint mapping, the feed-forward term for follower i is computed as

$$\mathbf{u}_i^{\mathrm{ff}}(t) = \mathrm{Adj}\big({}^{M_0}T_{M_i}(t)\big)\, \mathbf{u}_0(t). \tag{4.6}$$

The adjoint operator accounts for the coupling between translation and rotation when transforming twists between frames, thus providing a physically consistent velocity reference for the follower given the instantaneous leader-follower relative pose.

The final commanded planar velocity for follower i is obtained by superposition:

$$\mathbf{u}_i(t) = \mathbf{u}_i^{\mathrm{ff}}(t) + \mathbf{u}_i^{\mathrm{fb}}(t). \tag{4.7}$$

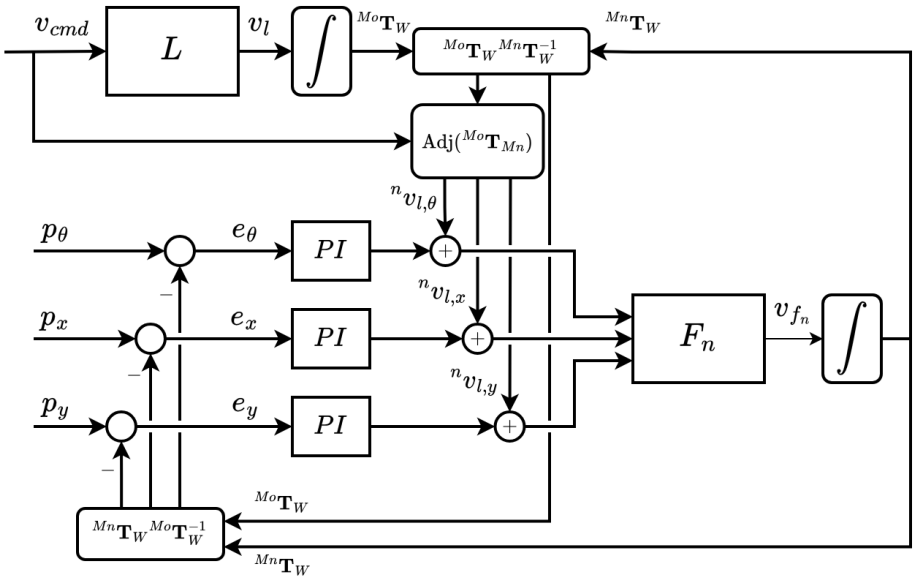


Figure 4.4 Control architecture. A reference velocity command is computed for the leader robot L. From the pose of the leader L and for every follower F_i, the relative transformation between the two is computed and fed to a PI controller plus feed-forward on the linear axes x, y and on the angular coordinate (yaw).

4.5 Compliance Control for Cooperative Manipulation

Even when the mobile bases maintain the desired formation, localization errors, communication delays, and actuation latencies can lead to discrepancies in the end-effector relative positioning. In cooperative transport, such discrepancies manifest as parasitic interaction forces and torques that may induce unwanted internal stresses on the payload and on the manipulators. A compliance strategy is therefore required to regulate contact wrenches and attenuate internal loading.

Let $\mathbf{w}_i = [f_{i,x}, f_{i,y}, f_{i,z}, \tau_{i,x}, \tau_{i,y}, \tau_{i,z}]^T \in \mathbb{R}^6$ be the wrench applied by the i-th end effector on the object, expressed in the i-th tool frame. Stacking all wrenches yields

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_0^T & \mathbf{w}_1^T & \dots & \mathbf{w}_n^T \end{bmatrix}^T \in \mathbb{R}^{6(n+1)}. \tag{4.8}$$

Assuming a rigid grasp and a payload frame P rigidly attached to the transported object, the resultant wrench on the payload in frame P is

$$\mathbf{w}_P = G\mathbf{w}, \tag{4.9}$$

where G is the grasp matrix:

$$G = \begin{bmatrix} \mathbb{I} & 0 & \dots & \mathbb{I} & 0 \\ S({}^{P}p_{E_0}) & \mathbb{I} & \dots & S({}^{P}p_{E_n}) & \mathbb{I} \end{bmatrix}. \tag{4.10}$$

Here ${}^{P}p_{E_i} \in \mathbb{R}^3$ is the translation from the i-th end-effector origin to the payload frame origin, expressed in P, and $S(\cdot)$ denotes the skew-symmetric matrix such that $S(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

Since the payload wrench space is six-dimensional, while the stacked wrench vector $\mathbf{w}$ lies in a higher-dimensional space for $n \geq 1$, the mapping in (4.9) is underdetermined and admits infinitely many solutions. A standard decomposition separates a particular solution that generates the desired payload wrench from a homogeneous component that lies in the null space of G:

$$\mathbf{w} = G^{\dagger}\mathbf{w}_P + \left(\mathbb{I} - G^{\dagger}G\right)\mathbf{w}_{\text{int}}, \tag{4.11}$$

where $G^{\dagger}$ is a (Moore-Penrose) pseudoinverse of G and $\left(\mathbb{I} - G^{\dagger}G\right)$ is the null-space projector. The vector $\mathbf{w}_{\text{int}}$ parameterizes internal forces and torques: components of the end-effector wrenches that do not affect the net wrench on the payload (i.e., they satisfy $G(\mathbb{I} - G^{\dagger}G)\mathbf{w}_{\text{int}} = \mathbf{0}$). These internal wrenches may cause excessive contact loading and structural stress; consequently, their attenuation is a primary control objective.

Explicit online computation and regulation of the null-space component in (4.11) is generally impractical in real robotic systems due to modeling uncertainties and sensing limitations. Instead, internal wrench regulation is commonly achieved indirectly through impedance control, which shapes the dynamic interaction between the manipulators and the payload.

To maintain consistency with the leader-follower paradigm adopted at the mobile base level, a master-slave compliance architecture is employed. The leader manipulator operates as a position-controlled master, defining the reference end-effector trajectory for the cooperative task. Each follower manipulator implements a Cartesian impedance control law with respect to this reference, thereby accommodating residual positioning errors while limiting interaction forces.

For follower i, the joint-space torque command is given by

$$\tau_i = \Lambda(q_i)\ddot{q}_i + \mathbf{C}(q_i, \dot{q}_i)\dot{q}_i + \mathbf{g}(q_i) + J_i(q_i)^T\left(K(\mathbf{x}_i - \mathbf{x}_r) + D\dot{\mathbf{x}}_i + M\ddot{\mathbf{x}}_i - \mathbf{w}_i\right), \tag{4.12}$$

where q_i, $\dot{q}_i$, and $\ddot{q}_i$ denote the joint position, velocity, and acceleration vectors of the i-th manipulator, respectively. The matrices $\Lambda(q_i)$, $\mathbf{C}(q_i, \dot{q}_i)$, and $\mathbf{g}(q_i)$ represent the manipulator

inertia matrix, Coriolis and centrifugal effects, and gravitational forces. The Jacobian $J_i(q_i)$ maps joint velocities to Cartesian end-effector velocities.

The vectors $\mathbf{x}_i$, $\dot{\mathbf{x}}_i$, and $\ddot{\mathbf{x}}_i$ denote the Cartesian position, velocity, and acceleration of the follower end effector, while $\mathbf{x}_r$ is the reference trajectory provided by the leader. The matrices K, D, and M are positive definite and diagonal, representing the desired stiffness, damping, and apparent mass of the Cartesian impedance, respectively. The wrench $\mathbf{w}_i$ is assumed to be measured directly at the end effector via a force/torque sensor.

4.6 Experiments

4.6.1 Simulations

The validation of the proposed costmap merging algorithm is conducted in two stages. First, it is tested using synthetically generated maps, and then it is extended to a physics-based simulation environment utilizing the Gazebo Sim simulator [108]. This approach is implemented alongside the follower control scheme described in previous sections. This two-stage validation strategy allows us to evaluate both the computational properties of the merging and planning algorithms, as well as their performance within a realistic robotic navigation framework.

All experiments are carried out with a formation consisting of two robots: a leader and a follower. This choice is based on the observation that the computational cost of the costmap merging procedure increases linearly with the number of robots in the formation. In contrast, the path computation time is determined solely by the size and internal structure of the three-dimensional costmap.

Costmap and Planning Computation Times

The initial experimental evaluation was conducted using synthetic maps that represented various navigation scenarios. These maps were generated using the occupancy map generator provided by Nvidia Isaac Sim and varied in layout and size to assess the scalability and computational performance of the proposed planning approach under different environmental conditions. For each scenario, both the dimensions of the map and the spatial distribution of obstacles were modified while maintaining a consistent resolution across all maps.

For each generated map, the initial navigation position of the leader robot was randomly selected within a region located at a distance of $0.1\max(MapHeight, MapWidth)$ from the map's origin, which is defined as the bottom-left pixel of the occupancy grid. Similarly,

the goal position was randomly sampled from the set of valid, obstacle-free cells located at a distance of $0.9\max(MapHeight, MapWidth)$ from the map's origin. This sampling strategy was employed to avoid trivial navigation problems and to ensure that the paths were sufficiently long and traversed a significant portion of the environment. The experiment was repeated ten times for each map, varying the start and goal configurations each time.

Path planning was performed using a modified implementation of the A* algorithm available within the ROS 2 Navigation stack. To operate on the three-dimensional weighted graph created by the costmap representation, the standard Manhattan heuristic was adjusted to account for the 26-connectivity of the grid. Specifically, the heuristic defined direction-dependent costs associated with all admissible neighboring transitions in the 3D lattice, ensuring consistency with the expanded neighborhood structure while preserving admissibility.

For each map, the planning algorithm was evaluated on two different computational platforms. An edge-computing configuration was implemented on an Nvidia Jetson Orin NX. In contrast, a workstation configuration ran on a high-performance system equipped with an Intel i9-13900HX CPU and an Nvidia RTX 4080 GPU. This comparison aimed to assess the computational burden of the proposed approach and its suitability for deployment on resource-constrained robotic platforms. The resulting performance metrics for both configurations are reported in Table 4.1.

MAP	Map Dimension [pixels]	Map Merging [s]	Path Computing[s]
$MAP1_j$	350x800	0.022 ± 0.011	0.438 ± 0.288
$MAP2_j$	500x900	0.045 ± 0.017	0.609 ± 0.321
$MAP3_j$	500x1500	0.081 ± 0.021	0.781 ± 0.433
$MAP4_j$	700x1500	0.102 ± 0.033	0.838 ± 0.489
$MAP1_w$	350x800	0.018 ± 0.008	0.111 ± 0.098
$MAP2_w$	500x900	0.032 ± 0.011	0.179 ± 0.087
$MAP3_w$	500x1500	0.041 ± 0.018	0.211 ± 0.101
$MAP4_w$	700x1500	0.066 ± 0.021	0.238 ± 0.129

Table 4.1 Map merging and path computing average time and standard deviation for ten repetitions on different map sizes, computed on a Jeston NX Orin directly on the leader AMR(MAP_j) and on a workstation(MAP_w).

Planning and Trajectory Tracking

The physical simulator Gazebo Sim [108] was used to model and simulate a formation of omnidirectional mobile robots. The simulations were carried out using the ROS 2 Nav2 navigation stack [78]. For the leader robot, velocity commands were generated using the default DWB controller [75] provided by Nav2. In contrast, the follower robots were controlled using the scheme depicted in Figure 4.4. The corresponding controller gains were tuned according to the Ziegler–Nichols method.

In this simulated scenario, relative localization among the mobile robots was estimated using the distributed approach proposed in [87]. This choice allows for the evaluation of the cooperative framework under realistic relative pose estimation conditions, in line with the assumptions made in the real-world experimental setup.

The simulation campaign included three different environments, all characterized by the same map dimensions. Figure 4.5 presents representative snapshots from the navigation phase of a single trial, showing the costmap at layer zero of the underlying three-dimensional costmap structure. Quantitative results are summarized in Table 4.2, which reports the average distance error between the followers and the leader, as well as the relative orientation error measured over the simulation runs.

MAP	Map Dimension [pixels]	Distance [m]	Angular [rad]
$MAP1_w$	350x800	0.019 ± 0.010	0.027 ± 0.009
$MAP2_w$	350x800	0.016 ± 0.008	0.023 ± 0.01
$MAP3_w$	350x800	0.020 ± 0.014	0.031 ± 0.012

Table 4.2 Errors from nominal leader-follower constraint, measured in average distance and relative angle. Computed over ten repetitions for each environment.

4.6.2 Real world experiments

Experimental Setup

The cooperative framework has then been tested on a real setup composed of two mobile custom mobile manipulators. The leader robot is built with an Omron LD60 differential mobile robot on top of which is mounted a Techman TM12 collaborative robot. An Intel NUC mini PC NUC11TNK is used to control both modules. The follower robot has a custom-made omnidirectional base with mecanum wheels and a UR10e collaborative manipulator with

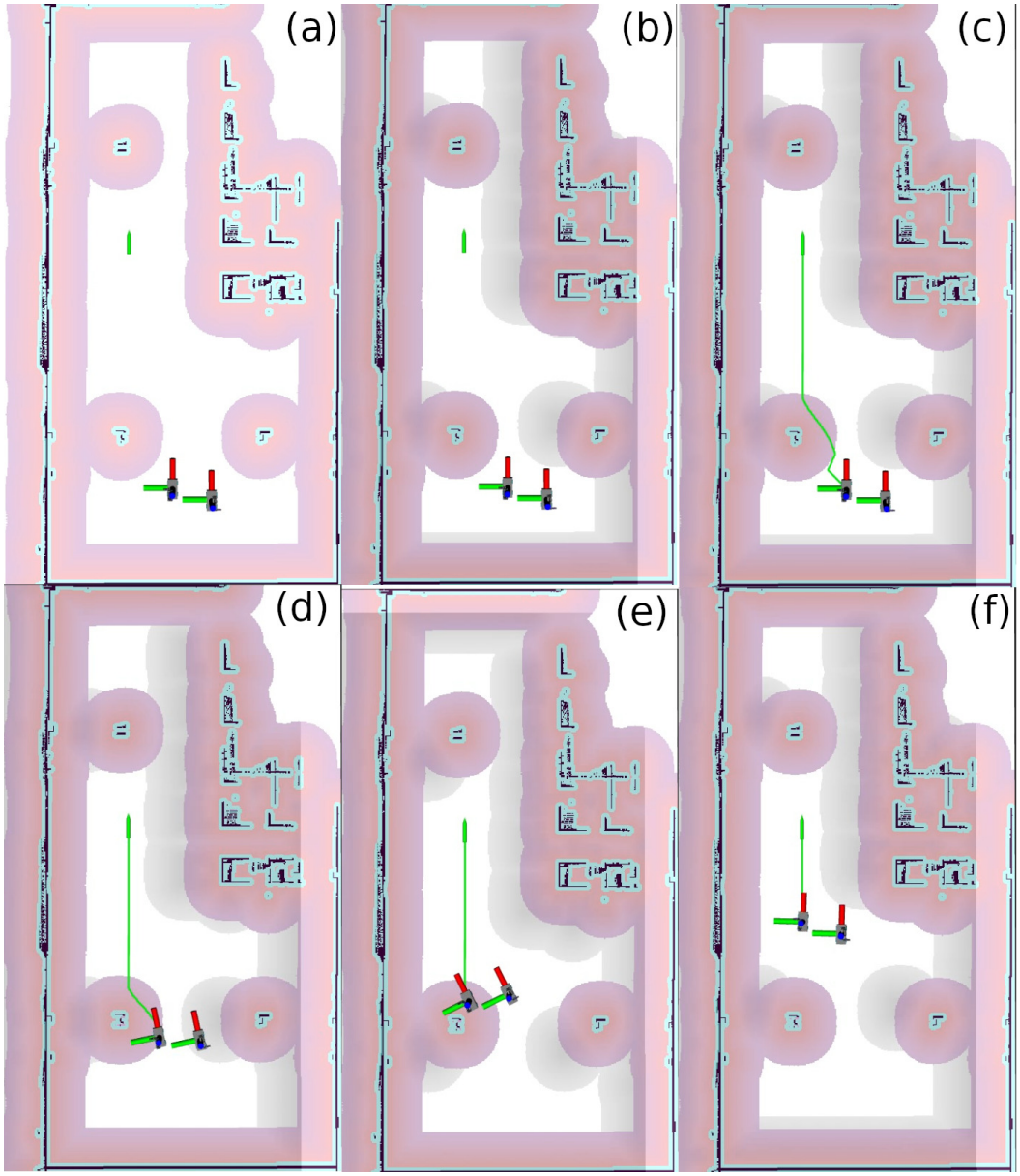


Figure 4.5 Navigation phases of the Gazebo simulation with a leader (left frame) and a follower (right frame): (a) A goal pose is set for the simulation, shown as green arrow; (b) The merged map is computed, only layer zero is shown (gray); (c) A Path is computed using A^*, shown in green; (d-f) The path is tracked by the robots. The inflated costmap, present in every figure, contains the objects on the map (black), the lethal cost area (cyan) and the exponentially decreasing inflated cost (red to blue).

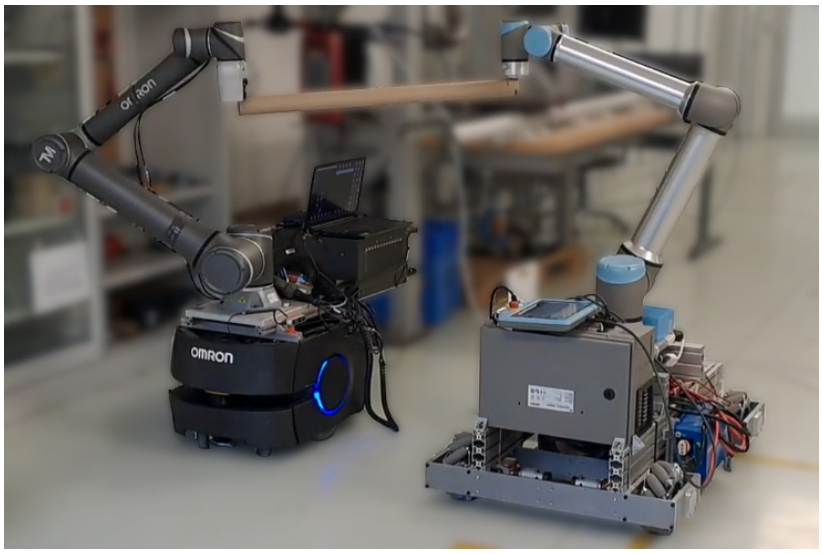


Figure 4.6 Experimental setup: On the left, the Omron/Techman leader of the formation; on the right the Custom/UR10e follower

an internal 6-axis force-torque sensor, both controlled by an Intel NUC NUC5i7RYH mini PC. The base is equipped with a 2D lidar module, used for localization. The high-level management of the formation is assigned to a remote workstation with an Intel i7-12700K CPU. Robots and workstation mount Ubuntu 22.04 operative system and rely on the ROS2 Humble framework for communication. The source code of the planner can be obtained at [88].

During the experimental tests, each mobile base is individually localized: the leader robot uses the Omron proprietary laser-based system for localization; the follower uses the Navigation2 [78] AMCL localization algorithm. The leader robot also mount the Regulated Pure Pursuit controller [77] from Nav2, to have an accurate tracking of the planned path.

Experiment Design and Results

The experimental scenario considered is illustrated in the first panel of Figure 4.10. The cooperative setup is required a robotic formation to transport a cardboard tube across the laboratory workspace. The payload is rigidly constrained by each robot along the three translational axes and two rotational axes. A limited degree of slack is deliberately introduced around the rotation about the Z-axis of the end effectors, in order to provide a small amount of mechanical compliance and to mitigate internal stresses within the formation.

A qualitative overview of the execution using the real hardware setup is provided in Figure 4.9, which reports a sequence of representative snapshots acquired during the experiment. The

same execution is shown in Figure 4.10 using the RViz visualization tool. This visualization allows inspection of the planned path at four selected time instants, together with the merged costmap employed by the planning module. It is important to note that the costmap displayed corresponds only to layer zero of the underlying three-dimensional costmap structure. In the figure, the standard costmap associated with the leader robot, depicted using magenta, cyan, and red colors according to the cost values, is overlaid on the merged costmap, which is shown in black. This representation is intended to highlight the spatial extension introduced by the merging process throughout the experiment. By observing the sequence of frames in Figure 4.10, it is possible to notice that the "shadow" generated by the costmap augmentation evolves throughout the execution of the task. This behavior directly reflects the deformation of the formation over time, which is captured by the time-varying relative transformation ${}^{M_0}T_{M_1}(t)$ between the leader and the follower frames. As the relative pose between the robots changes, the merged costmap is correspondingly updated, resulting in a dynamic modification of the perceived occupied regions for the formation.

An additional aspect of interest can be observed in the last two panels of Figures 4.9 and 4.10. In these instances, the follower robot is shown to traverse regions associated with high cost values in the merged costmap without experiencing any physical collision with obstacles. This behavior is consistent with the adopted representation, as the merged costmap encodes the environment exclusively from the leader's viewpoint and therefore it is not correlated to collision constraints for the follower robots. Consequently, as long as the leader does not enter regions characterized by high cost values in the merged costmap, the overall configuration of the formation remains safe and feasible.

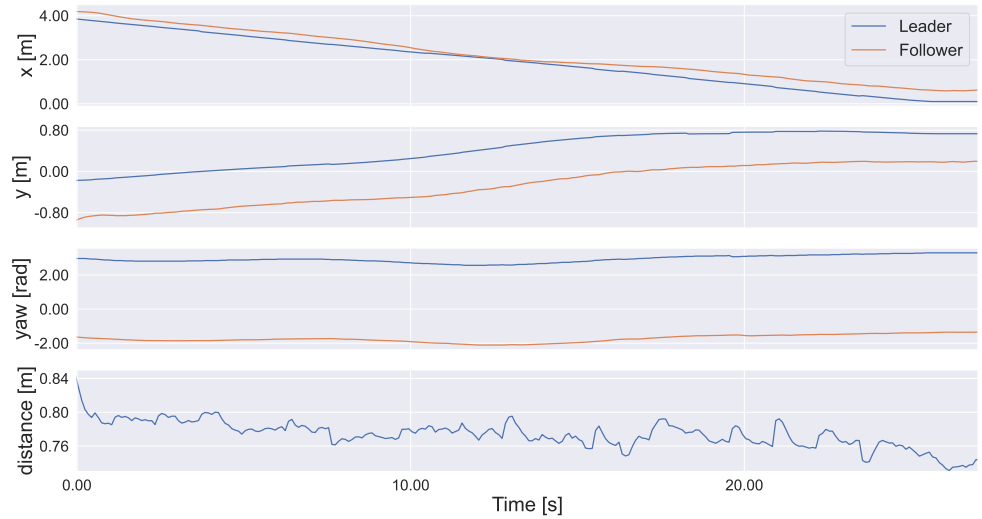


Figure 4.7 Trajectory followed by leader and follower mobile bases, with respect to global map frame. Bottom plot shows the Euclidian distance between the two bases.

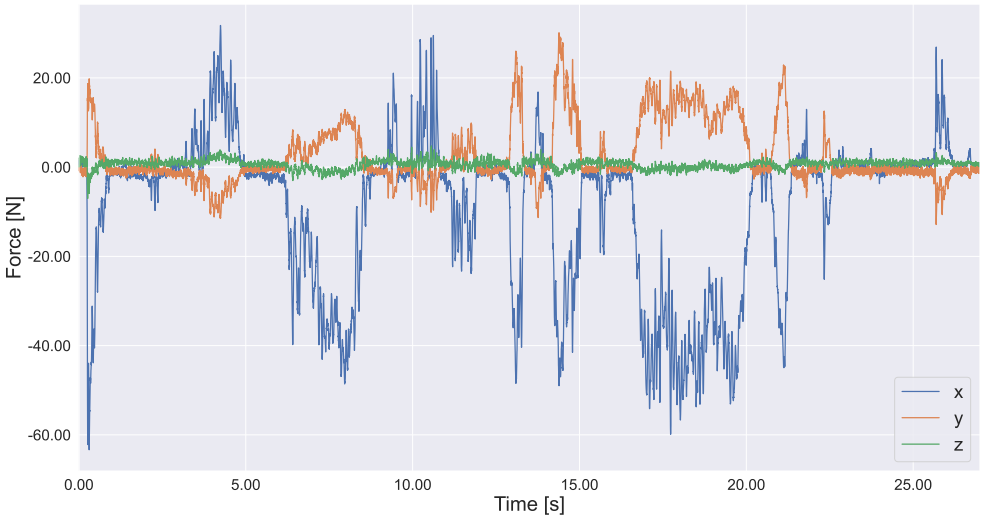


Figure 4.8 Force measured at the follower end , w.r.t. end effector frame: x is aligned with the main direction of the formation and z is vertical downward

A quantitative analysis of the motion of the formation is presented in Figure 4.7, which reports the trajectories of both the leader and the follower expressed in the global map frame. In both the x and y directions, the follower is shown to closely reproduce the trajectory of the leader. The most noticeable discrepancies occur at the beginning of the motion, where the leader's localization algorithm performs a minor instantaneous correction that propagates to the follower through the cooperative control scheme, and during segments characterized by rapid variations in trajectory curvature. The tracking performance is also effective in terms of heading. The constant offset observed in the heading profiles is attributable to the different orientations of the leader and follower base reference frames. The effects of base motion tracking on the constrained manipulators are further illustrated in Figure 4.8, which reports the forces measured at the follower's end effector during task execution. To further assess the robustness of the proposed cooperative framework and to enable a meaningful comparison with the results obtained in simulation, ten additional experimental trials were conducted. In accordance with the strategy adopted in the simulated case, as reported in Table 4.1, the global planning procedure was executed on an external workstation to reduce the computational load on the leader robot. Only the resulting planned path was transmitted to the leader for execution.

The results of these experiments are summarized in Table 4.3, which reports the average error in the nominal relative positioning of the follower with respect to the leader, together with the corresponding standard deviation, as well as the average costmap merging time. Despite the increased size of the map employed in the real-world experiments compared to the simulated scenario, the processing time required for costmap merging remains comparable to that observed in simulation. With respect to the relative positioning accuracy, an increase in the average distance error is observed, which can be attributed to realistic effects such as wheel slippage and unmodeled disturbances that impact localization accuracy. Nevertheless, the observed standard deviation is consistent with the trends reported in simulation results (Table 4.2), thereby confirming the effectiveness and robustness of the proposed cooperative framework when deployed on real hardware.

Map size [pixel]	Distance [m]	Angular [rad]	Merging time [s]
1400x1500	0.048 ± 0.010	-0.001 ± 0.0014	0.3065 ± 0.0065

Table 4.3 Localization error from nominal leader-follower constraint, measured in average value and standard deviation on distance and relative angle, and costmap merging time over ten experiments

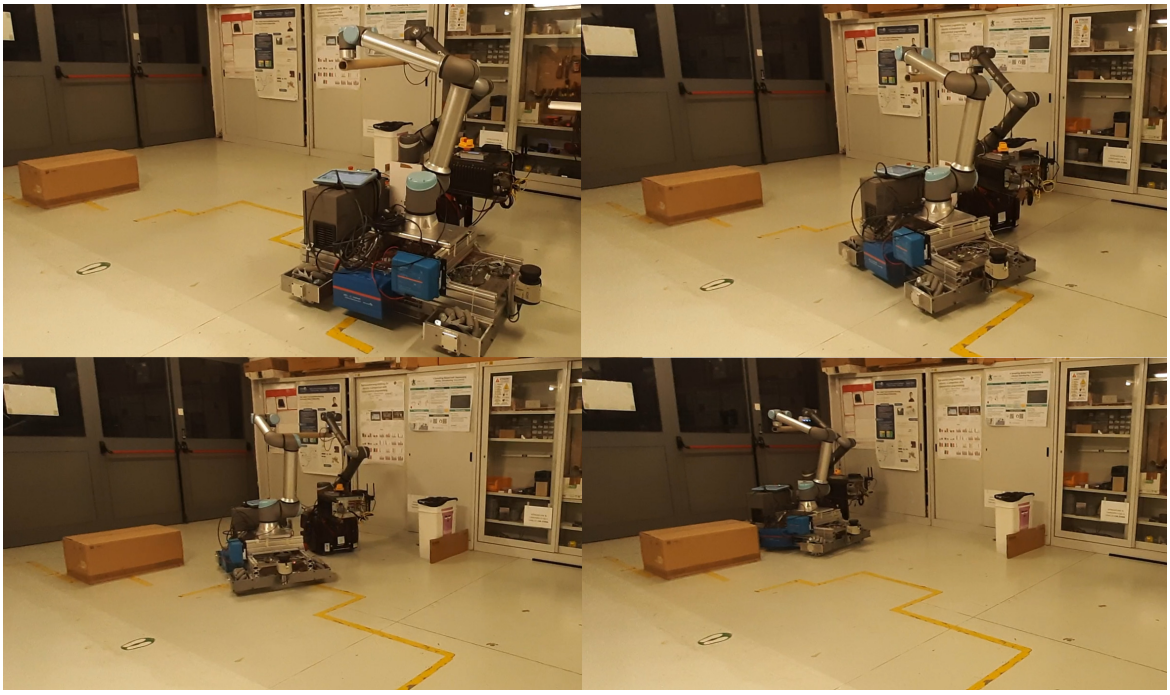


Figure 4.9 Snapshots of multi-robot cooperative transport experiments

Figure 4.10 Visualization of multi-robot cooperative transport using RVIz software. The most left panel shows the standard costmap computed by the leader robot. The remaining four panels show the execution of the test and the layer zero of the merged costmap

4.6.3 Discussion

The proposed cooperative framework has several aspects that require careful consideration during both its setup and deployment. One of the most critical factors is the accuracy and stability of the localization process for each robot in the formation. Both the leader's planning strategy, implemented through the costmap merging procedure, and the follower control law depend on the relative transformation ${}^{M_0}T_{M_i}$. When this transformation is derived from independent localization estimates, it is essential to avoid abrupt corrections or oscillations in the estimated poses. Such effects can directly affect the formation maintenance mechanism, potentially leading to unstable or degraded behavior. As an alternative, direct relative localization approaches, such as using visual or fiducial markers, or estimation strategies as proposed in [87], can significantly enhance estimation accuracy by reducing the uncertainties that arise from multiple independent localization processes.

The overall performance of the cooperative framework and the robustness of the localization process are closely tied to the quality of the costmaps generated by each robot. Since the augmented costmap is created by combining multiple local maps, it can be affected by mapping artifacts that may occur during the SLAM process, as well as by sensor noise, which can introduce false or misleading obstacles. These artifacts can directly propagate to the formation-level representation of the environment, severely compromising the planning procedure and leading to overly conservative or infeasible trajectories. Therefore, effective sensor filtering and post-processing techniques are essential to ensure that consistent and reliable costmaps are generated.

Another important aspect to address is the environment perceived by each robot within the formation. If a robot detects other members of the formation with its onboard sensors, these detections may be incorrectly integrated into its local costmap and subsequently into the leader's augmented costmap, potentially introducing fake obstacles. To prevent self-induced map corruption, this issue may be mitigated by differentiating sensor measurements across multiple costmaps, explicitly recognizing other robots in the sensor data, or, when possible, avoiding the scan areas known to be occupied by the formation.

4.7 Summary

A comprehensive leader-follower framework is introduced to facilitate cooperative transport with teams of mobile manipulators operating in dynamic and potentially unstructured environments. The overall strategy combines centralized planning with decentralized execution,

addressing two interrelated requirements: generating globally feasible motions for the team while adhering to the rigid geometric constraints imposed by a shared payload.

A key contribution of this work is an environment representation and planning pipeline that merges the local navigation costmaps produced by each robot into a formation-aware model. Instead of relying solely on a single robot's perception, the approach integrates distributed obstacle information into a unified representation that supports global path planning for the group. Formation feasibility is maintained by explicitly accounting for the motion of the rigid formation during both translation and rotation. This is achieved through a layered representation that encodes collision-free configuration space across various discretized formation orientations. Standard graph-based planners can then compute paths that conform to formation constraints. Periodic replanning is implemented to adapt to environmental changes and discard outdated plans. The maximum replanning frequency is limited by merging and planning times and thus depends on the size of the environment map and on the number of formation partecipants.

A leader-follower control architecture is employed to execute the planned motion. Path tracking is assigned to the leader using established navigation controllers appropriate for mobile bases. Meanwhile, each follower maintains the desired relative pose with respect to the leader through a decentralized feedback strategy, complemented by a feed-forward term derived from the leader's motion command.

Cooperative manipulation aspects are addressed through a compliance layer designed to alleviate internal stresses caused by localization drift, communication latency, and actuation discrepancies.

Validation is conducted through both simulation and real-world experiments with two mobile manipulators transporting a rigid object. Simulations assess the scalability and computational characteristics of the merged planning representation and quantify formation tracking accuracy across multiple environments. Real-world experiments demonstrate the practical feasibility of the framework under realistic localization conditions. The results show that the framework maintains acceptable formation errors while ensuring manageable planning and map-merging runtimes.

Chapter 5

Conclusions

5.1 Summary

This thesis is situated within the broader effort to move robotics from highly structured, isolated automation toward systems that operate reliably in open, dynamic, and contact-rich environments. In this setting, physical interaction is not an exceptional condition but an expected operating mode: robots must accommodate uncertainty, sustain stable contact, and remain predictable while adapting their motion in response to external forces, often originating from human partners or from task-driven constraints.

The contributions of the thesis can be summarized as follows:

- **Elasto-plastic interaction control for stiff, position-controlled manipulators:** a control framework inspired by elasto-plastic behavior that blends reversible deformation for small perturbations with controlled, permanent offsets under sustained interaction, enabling intuitive guidance while preserving stability and repeatability.
- **Whole-body extension to mobile manipulation via hierarchical optimization:** an HQP-based formulation that embeds elasto-plastic task behavior within a whole-body controller, coordinating base and arm motion to realize compliant interaction while maintaining task priorities and feasibility across the kinematic chain.
- **Trajectory recovery after interaction-driven deviations:** an approach to rejoin the nominal motion after permanent offsets by introducing and optimizing intermediate references within the control hierarchy, enabling controlled transition back to planned execution without abrupt discontinuities.

- **Formation-aware planning for cooperative transport:** a navigation strategy that accounts for rigid payload constraints through path generation on formation-aware costmaps and a practical formation controller based on relative-pose regulation with leader-velocity feed-forward, supporting cooperative motion under realistic communication and estimation limitations.
- **Integrated experimental validation:** ROS-compatible architectures and real-platform evaluations that demonstrate the feasibility of the proposed methods in representative interaction-centric scenarios.

5.2 Future Works

The results of this thesis offer a starting point for multiple future studies and in-depth explorations that can build on its contributions. These includes:

- Elastoplastic dual-manipulation: Actual formulation of the elastoplastic controller is limited to single-arm mobile manipulators. By considering dual-arm manipulators, better control of the transported object, especially on orientation, could be achieved as well as more human-like transport behaviors.
- Deformable payload transport: cooperative manipulation through elastoplastic control is limited to transport of rigid objects due to the required force-torque feedback. Carriage of flexible payload could be tackled by introducing vision-based systems in order to estimate the deformation, either indirectly by skeleton tracking or directly through images of the object deformation.
- Elastoplastic multi-robot formation: elastoplastic logic can be interpreted as a leader-follower arbitration system. Multi-robot formations could leverage the elastoplastic method to regulate roles in a leader-follower formation scheme, allowing leader variation when the actual is unable to react to obstacles and disturbances.
- Augmented costmap approach for real-time planning: Augmented costmap approach is applied to global path planning. The same augmenting methodology could be applied to generate local costmap on which operate with real-time planners, allowing the leader to perform dynamic obstacle avoidance for the whole formation.

List of Publications

List of publications relevant for the presented work and as result of collaborations in the field of control and robotics.

- Venezia, A.*, Romata, I.*, Nicola, G.*, **Fausti, R.***, Nitti, M.*, Liso, A.*, Renò, V., Pedrocchi, N., Beschi, M., Longo, N., Di Stefano, G., Panicucci, S. (2024). Enabling Cognitive Robotics Through Autonomous Motion Planning and Model-Driven Intuitive Programming. *In: Secchi, C., Marconi, L. European Robotics Forum 2024. ERF 2024. Springer Proceedings in Advanced Robotics, vol 32. Springer*

- **Fausti, R.**, Ghidini, S., Beschi, M., Pedrocchi, N. (2024) Elasto-plastic Control for Physical Human-Robot Interaction. *29th IEEE International Conference on Emerging Technologies and Factory Automation.*

- Mutti, S., **Fausti, R.**, Pedrocchi, N., Valente, A. (2025) Augmented Costmap-Based Path Planning and Control for Multi-Mobile Robot Rigid Formation, *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*.

- Ferrari, M., **Fausti, R.**, Tonola, C., Delledonne, M., Sandrini, S., Beschi, M., & Villagrossi, E. (2025). A distributed control architecture for a multi-agent robotic cell: a battery pack disassembly case study. *International Journal of Computer Integrated Manufacturing*

References

[1] ISO/TC 299/WG 3. Robotics — safety requirements. part 1: Industrial robots. Standard, International Organization for Standardization, 2025.

[2] ISO/TC 299/WG 3. Robotics — safety requirements. part 2: Industrial robot applications and robot cells. Standard, International Organization for Standardization, 2025.

[3] Fares J. Abu-Dakka and Matteo Saveriano. Variable impedance control and learning—a review. *Frontiers in Robotics and AI*, Volume 7 - 2020, 2020. ISSN 2296-9144. doi: 10.3389/frobt.2020.590681.

[4] Nour Abujabal, Raouf Fareh, Saif Sinan, Mohammed Baziyad, and Maamar Bettayeb. A comprehensive review of the latest path planning developments for multi-robot formation systems. *Robotica*, 41(7):2079–2104, 2023. doi: 10.1017/S0263574723000322.

[5] Richard J Adams and Blake Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on robotics and Automation*, 15(3):465–474, 2002.

[6] Don Joven Agravante, Andrea Cherubini, Antoine Bussy, Pierre Gergondet, and Abderrahmane Kheddar. Collaborative human-humanoid carrying using vision and haptic sensing. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 607–612, 2014. doi: 10.1109/ICRA.2014.6906917.

[7] Hyo-Sung Ahn. *Formation Control*. Springer Cham, 2020.

[8] Javier Alonso-Mora, Stuart Baker, and Daniela Rus. Multi-robot formation control and object transport in dynamic environments via constrained optimization. *The International Journal of Robotics Research*, 36(9):1000–1021, 2017.

[9] Tim Bailey and Hugh Durrant-Whyte. Slam—part i: The essential algorithms. *IEEE Robotics & Automation Magazine*, 13(2):108–117, 2006. doi: 10.1109/MRA.2006.1638022.

[10] Tucker Balch and Ronald C Arkin. Behavior-based formation control for multirobot teams. *IEEE transactions on robotics and automation*, 14(6):926–939, 2002.

[11] R.W. Beard, J. Lawton, and F.Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, 2001. doi: 10.1109/87.960341.

[12] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002. ISSN 0921-8890. Ninth International Symposium on Intelligent Robotic Systems.

[13] Ahmed Benzerrouk, Lounis Adouane, Laurent Lequievre, and Philippe Martinet. Navigation of multi-robot formation in unstructured environment using dynamical virtual structures. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5589–5594. IEEE, 2010.

[14] F. Benzi, C. Mancus, and C. Secchi. Whole-body control of a mobile manipulator for passive collaborative transportation. In *IFAC-PapersOnLine*, volume 55, pages 106–112. Elsevier B.V., 10 2022. ISBN 9781713867890.

[15] Manuel Beschi, Nicola Pedrocchi, and Stefano Ghidini. cnr_ros_control, 2025. URL https://github.com/CNR-STIIMA-IRAS/cnr_ros_control.

[16] Khadir Lakhdar Besseghieur, Radosław Trębiński, Wojciech Kaczmarek, and Jarosław Panasiuk. From trajectory tracking control to leader–follower formation control. *Cybernetics and Systems*, 51(4):339–356, 2020.

[17] Simon Bøgh, Mads Hvilshøj, Morten Kristiansen, and Ole Madsen. Autonomous industrial mobile manipulation (aimm): From research to industry. In *Proceedings of the 42nd International Symposium on Robotics*. VDE Verlag GMBH, 2011.

[18] Suraj Borate, Rwik Rana, Praveen Venkatesh, and Madhu Vadali. Ff-rrt*: A sampling-based planner for multirobot global formation path planning. *Journal of Mechanisms and Robotics*, 16(10):104502, 09 2024. ISSN 1942-4302.

[19] Gabriele Buondonno. Eiquadprog. https://github.com/stack-of-tasks/eiquadprog, 2025.

[20] Ben Burgess-Limerick, Jesse Haviland, Chris Lehnert, and Peter Corke. Reactive base control for on-the-move mobile manipulation in dynamic environments. *IEEE Robotics and Automation Letters*, 9(3):2048–2055, 2024.

[21] Federico Califano, Ramy Rashad, Cristian Secchi, and Stefano Stramigioli. On the use of energy tanks for robotic systems. In Pablo Borja, Cosimo Della Santina, Luka Peternel, and Elena Torta, editors, *Human-Friendly Robotics 2022*, pages 174–188, Cham, 2023. Springer International Publishing. ISBN 978-3-031-22731-8.

[22] C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3): 419–425, 1995.

[23] Bingjie Chen, Yancong Wei, Rihao Liu, Chenxi Han, Houde Liu, Chongkun Xia, Liang Han, and Bin Liang. Safe expeditious whole-body control of mobile manipulators for collision avoidance. *arXiv preprint arXiv:2409.14775*, 2024.

[24] Lei Chen and Ma Baoli. A nonlinear formation control of wheeled mobile robots with virtual structure approach. In *2015 34th Chinese Control Conference (CCC)*, pages 1080–1085. IEEE, 2015.

[25] YongBo Chen, JianQiao Yu, XiaoLong Su, and GuanChen Luo. Path planning for multi-uav formation. *Journal of Intelligent & Robotic Systems*, 77(1):229–246, 2015.

[26] Vishnu R. Desaraju and Jonathan P. How. Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In *2011 IEEE International Conference on Robotics and Automation*, pages 4956–4961, 2011. doi: 10.1109/ICRA.2011.5980392.

[27] Alexander Dietrich, Xuwei Wu, Kristin Bussmann, Christian Ott, Alin Albu-Schffer, and Stefano Stramigioli. Passive hierarchical impedance control via energy tanks. *IEEE Robotics and Automation Letters*, 2:522–529, 4 2017. ISSN 23773766.

[28] Gamini Dissanayake, Paul Newman, Steven Clark, Hugh Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001. doi: 10.1109/70.938381.

[29] Vincent Duchaine and Clement M. Gosselin. General model of human-robot cooperation using a novel velocity based variable impedance control. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 446–451, 2007. doi: 10.1109/WHC.2007.59.

[30] P. Dupont, Brian Armstrong, and Vincent Hayward. Elasto-plastic friction model: Contact compliance and stiction. *Proceedings of the American Control Conference*, 2: 1072–1077, 2000.

[31] Boston Dynamics. Stretch, 2025. URL https://bostondynamics.com/products/stretch/.

[32] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. doi: 10.1109/2.30720.

[33] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33:1006–1028, 2014.

[34] Abdul Qadir Faridi, Sanjeev Sharma, Anupam Shukla, Ritu Tiwari, and Joydip Dhar. Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment. *Intelligent Service Robotics*, 11(2): 171–186, 2018.

[35] Roberto Fausti. Elastoplastic controller. https://github.com/JRL-CARI-CNR-UNIBS/elastoplastic_phri_controller/tree/etfa-2024, 2024.

[36] Roberto Fausti. Elastoplastic controller for mobile manipulators. https://github.com/JRL-CARI-CNR-UNIBS/elastoplastic_phri_controller/tree/ros2, 2025.

[37] Roberto Fausti, Stefano Ghidini, Manuel Beschi, and Nicola Pedrocchi. Elastoplastic control for physical human-robot interaction. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 01–07. IEEE, 2024.

[38] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4948–4953, 2013. ISSN 10504729.

[39] Federica Ferraguti, Nicola Preda, Auralius Manurung, Marcello Bonfe, Olivier Lambercy, Roger Gassert, Riccardo Muradore, Paolo Fiorini, and Cristian Secchi. An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31:1073–1088, 10 2015. ISSN 15523098.

[40] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. Variable impedance control of redundant manipulators for intuitive human-robot physical interaction. *IEEE Transactions on Robotics*, 31:850–863, 8 2015. ISSN 15523098.

[41] Paolo Franceschi, Nicola Pedrocchi, and Manuel Beschi. Human-robot role arbitration via differential game theory. *IEEE Transactions on Automation Science and Engineering*, 21(4):5953–5968, 2024. doi: 10.1109/TASE.2023.3320708.

[42] Paolo Franceschi, Davide Cassinelli, Nicola Pedrocchi, Manuel Beschi, and Paolo Rocco. Design of an assistive controller for physical human–robot interaction based on cooperative game theory and human intention estimation. *IEEE Transactions on Automation Science and Engineering*, 22:5741–5756, 2025. doi: 10.1109/TASE.2024.3429643.

[43] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[44] Stefano Ghidini. *Design and Development of Control Strategies to Facilitate Human-Robot Interaction*. PhD thesis, Università degli studi di Brescia, 2023.

[45] Stavros Grafakos, Fotios Dimeas, and Nikos Aspragathos. Variable admittance control in phri using emg-based arm muscles co-activation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 001900–001905, 2016. doi: 10.1109/SMC.2016.7844516.

[46] Nacer Hacene and Boubekeur Mendil. Behavior-based autonomous navigation and formation control of mobile robots in unknown cluttered dynamic environments with dynamic target tracking. *International Journal of Automation and Computing*, 18(5): 766–786, 2021.

[47] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

[48] Shude He, Min Wang, Shi-Lu Dai, and Fei Luo. Leader–follower formation control of usvs with prescribed performance and collision avoidance. *IEEE transactions on industrial informatics*, 15(1):572–581, 2018.

[49] Adam Heins, Michael Jakob, and Angela P. Schoellig. Mobile manipulation in unknown environments with differential inverse kinematics control. In *2021 18th Conference on Robots and Vision (CRV)*, pages 64–71, 2021. doi: 10.1109/CRV52889.2021.00017.

[50] Neville Hogan. Impedance control: An approach to manipulation: Part i-theory. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 107:1–7, 3 1985. ISSN 15289028.

[51] Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Causal policy gradient for whole-body mobile manipulation. *arXiv preprint arXiv:2305.04866*, 2023.

[52] Ryojun Ikeura and Hikaru Inooka. Variable impedance control of a robot for cooperation with a human. *Proceedings - IEEE International Conference on Robotics and Automation*, 3:3097–3102, 1995. ISSN 10504729.

[53] Konstantinos Ioannidis, Georgios Ch Sirakoulis, and Ioannis Andreadis. A path planning method based on cellular automata for cooperative robots. *Applied Artificial Intelligence*, 25(8):721–745, 2011.

[54] Alexander Jahn, Reza Javanmard Alitappeh, David Saldaña, Luciano CA Pimenta, Andre G Santos, and Mario FM Campos. Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278. IEEE, 2017.

[55] Gitae Kang, Hyun Seok Oh, Joon Kyue Seo, Uikyum Kim, and Hyouk Ryeol Choi. Variable admittance control of robot manipulators based on human intention. *IEEE/ASME Transactions on Mechatronics*, 24(3):1023–1032, 2019. doi: 10.1109/TMECH.2019.2910237.

[56] Sanghyun Kim, Keunwoo Jang, Suhan Park, Yisoo Lee, Sang Yup Lee, and Jaeheung Park. Whole-body control of non-holonomic mobile manipulator based on hierarchical quadratic programming and continuous task transition. In *2019 IEEE 4TH International Conference on Advanced Robotics and Mechatronics (ICARM 2019)*, pages 414–419, 2019.

[57] Stephen Kloder and Seth Hutchinson. Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics*, 22(4):650–665, 2006.

[58] Kazuho Kobayashi, Seiya Ueno, and Takehiro Higuchi. Multi-robot patrol algorithm with distributed coordination and consciousness of the base station's situation awareness. *arXiv preprint arXiv:2307.08966*, 2023.

[59] Christopher M Korpela, Todd W Danko, and Paul Y Oh. Mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 65(1):93–101, 2012.

[60] Kazuhiro Kosuge and Norihide Kazamura. Control of a robot handling an object in cooperation with a human. In *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI*, pages 142–147. IEEE, 1997.

[61] Thomas Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013. doi: 10.1016/j.robot.2013.07.007.

[62] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[63] Przemyslaw A Lasota, Terrence E Fong, and Julie A Shah. A survey of methods for safe human-robot interaction. *Foundations and Trends in Robotics*, 5(4):261–349, 2017. doi: 10.1561/2300000052.

[64] Bo-Nian Lee, Hua-Hsuan Yeh, and Yen-Chen Liu. Whole-body impedance control for mobile manipulators with online uncertainty compensation. In *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 626–631, 2024. doi: 10.1109/ROBIO64047.2024.10907740.

[65] Giroung Lee and Dongkyoung Chwa. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intelligent Service Robotics*, 11(1): 127–138, 2018.

[66] M Anthony Lewis and Kar-Han Tan. High precision formation control of mobile robots using virtual structures. *Autonomous robots*, 4(4):387–403, 1997.

[67] Chao Li, Zhi Zhang, Guihua Xia, Xinru Xie, and Qidan Zhu. Efficient force control learning system for industrial robots based on variable impedance control. *Sensors*, 18(8):2539, 2018.

[68] Haijian Li, Weizhe Meng, ZongLin Han, Zihan Zhang, and Yanfang Yang. Vehicle platoon in road traffic: A survey of modeling, communication, controlling and perspectives. *Physica A: Statistical Mechanics and its Applications*, 674:130757, 2025. ISSN 0378-4371.

[69] Jun Li, Haibo Gao, Yuhui Wan, Joseph Humphreys, Christopher Peers, Haitao Yu, and Chengxu Zhou. Whole-body control for a torque-controlled legged mobile manipulator. In *Actuators*, volume 11, page 304. MDPI, 2022.

[70] Keqiang Li, Jiawei Wang, and Yang Zheng. Cooperative formation of autonomous vehicles in mixed traffic flow: Beyond platooning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15951–15966, 2022.

[71] Daqing Lu, Eitan Marder-Eppstein, and Francisco Allen. Layered costmaps for context-sensitive navigation. In *Proceedings of the 7th International Conference on Human-Robot Interaction*, pages 485–492, 2014. doi: 10.1145/2559636.2559666.

[72] Xiaobin Lu, MM Khonsari, and ERM Gelinck. The stribeck curve: experimental results and theoretical prediction. *Journal of tribology*, 128(4):789–794, 2006.

[73] Yingchong Ma, Gang Zheng, and Wilfrid Perruquetti. Cooperative path planning for mobile robots based on visibility graph. In *Proceedings of the 32nd Chinese Control Conference*, pages 4915–4920, 2013.

[74] Yuntao Ma, Farbod Farshidian, Takahiro Miki, Joonho Lee, and Marco Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 7(2):2377–2384, 2022. doi: 10.1109/LRA.2022.3143567.

[75] S. Macenski, T. Moore, DV Lu, A. Merzlyakov, and M. Ferguson. From the desks of ros maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2. *Robotics and Autonomous Systems*, 2023.

[76] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2718–2725, 2020. doi: 10.1109/IROS45743.2020.9341207.

[77] Steve Macenski, Shrijit Singh, Francisco Martín, and Jonatan Ginés. Regulated pure pursuit for robot path tracking. *Autonomous Robots*, 47(6):685–694, 2023.

[78] Steven Macenski, Francisco Martin, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[79] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[80] Al Jaber Mahmud, Amir Hossain Raj, Duc M Nguyen, Weizi Li, Xuesu Xiao, and Xuan Wang. Human-robot co-transportation using disturbance-aware mpc with pose optimization. *arXiv preprint arXiv:2404.00514*, 2024.

[81] Abhijit Makhal and Alex K Goins. Reuleaux: robot base placement by reachability analysis. In *2018 second IEEE international conference on robotic computing (IRC)*, pages 137–142. IEEE, 2018.

[82] Eitan Marder-Eppstein, Chris Berger, Tully Foote, Brian Gerkey, and Kurt Konolige. Office marathon: Robust navigation in an indoor office environment. In *2010 IEEE International Conference on Robotics and Automation*, pages 300–307. IEEE, 2010. doi: 10.1109/ROBOT.2010.5509693.

[83] F. Matoui, B. Boussaid, B. Metoui, G.B. Frej, and M.N. Abdelkrim. Path planning of a group of robots with potential field approach: decentralized architecture. *IFAC-PapersOnLine*, 50(1):11473–11478, 2017. ISSN 2405-8963. 20th IFAC World Congress.

[84] Fethi Metoui, Boumedyen Boussaid, and Mohamed Naceur Abdelkrim. Path planning for a multi-robot system with decentralized control architecture. In *New Trends in Robot Control*, pages 229–259. Springer, 2020.

[85] Maria Vittoria Minniti, Farbod Farshidian, Ruben Grandia, and Marco Hutter. Whole-body mpc for a dynamically stable mobile manipulator. *IEEE Robotics and Automation Letters*, 4(4):3687–3694, 2019. doi: 10.1109/LRA.2019.2927955.

[86] Hans P Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, pages 116–121. IEEE, 1985. doi: 10.1109/ROBOT.1985.1087316.

[87] Stefano Mutti and Giovanni Dimauro. Distributed lidar based control for cooperative transportation with multiple autonomous mobile robots. In *2023 IEEE International Conference on Industrial Technology (ICIT)*, pages 1–5. IEEE, 2023.

[88] Stefano Mutti and Roberto Fausti. Multi-robot collaborative framework. https://github.com/JRL-CARI-CNR-UNIBS/nav2_cooperative_planner, 2025.

[89] Stefano Mutti, Roberto Fausti, Nicola Pedrocchi, and Anna Valente. Augmented costmap-based path planning and control for multi-mobile robot rigid formation. In *2025 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 1–6. IEEE, 2025.

[90] Dharna Nar and Radhika Kotecha. Optimal waypoint assignment for designing drone light show formations. *Results in Control and Optimization*, 9:100174, 2022. ISSN 2666-7207.

[91] Alex Nash, Sven Daniel, Sven Koenig, and Ariel Felner. Theta*: Any-angle path planning on grids. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI)*, pages 1177–1183, 2007. doi: 10.1613/jair.2994.

[92] Giorgio Nicola, Stefano Mutti, Enrico Villagrossi, and Nicola Pedrocchi. Depth image-based deformation estimation of deformable objects for collaborative mobile transportation. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 2658–2664. IEEE, 2023.

[93] Jingmou Nie, Yaonan Wang, Yang Mo, Zhiqiang Miao, Yiming Jiang, Hang Zhong, and Jie Lin. An hqp-based obstacle avoidance control scheme for redundant mobile manipulators under multiple constraints. *IEEE Transactions on Industrial Electronics*, 70(6):6004–6016, 2023. doi: 10.1109/TIE.2022.3196390.

[94] Kohei Nozaki and Toshiyuki Murakami. A motion control of two-wheels driven mobile manipulator for human-robot cooperative transportation. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 1574–1579. IEEE, 2009.

[95] Ernesto Olguín-Díaz, Gustavo Arechavaleta, Gerardo Jarquín, and Vicente Parra-Vega. A passivity-based model-free force–motion control of underwater vehicle-manipulator systems. *IEEE Transactions on Robotics*, 29(6):1469–1484, 2013. doi: 10.1109/TRO.2013.2277535.

[96] Omron. Omron mobile manipulator (moma), 2025. URL https://robotics.omron.com/products/mobile-robots/mobile-manipulator/.

[97] Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. Unified impedance and admittance control. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 554–561, 2010. ISSN 10504729.

[98] Michael Otte and Nikolaus Correll. Any-com multi-robot path-planning with dynamic teams: Multi-robot coordination under communication constraints. In *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, pages 743–757. Springer, 2014.

[99] Jinseong Park, Yong-Sik Shin, and Sanghyun Kim. Object-aware impedance control for human–robot collaborative task with online object parameter estimation. *IEEE Transactions on Automation Science and Engineering*, 22:8081–8094, 2025. doi: 10.1109/TASE.2024.3477471.

[100] Riccardo Parosi, Mattia Risiglione, Darwin G. Caldwell, Claudio Semini, and Victor Barasuol. Kinematically-decoupled impedance control for fast object visual servoing and grasping on quadruped manipulators. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2023. doi: 10.1109/IROS55552.2023.10341714.

[101] Luka Peternel, Nikos Tsagarakis, Darwin Caldwell, and Arash Ajoudani. Adaptation of robot physical behaviour to human fatigue in human-robot co-manipulation. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 489–494, 2016.

[102] Carlos Plou, Pablo Pueyo, Ruben Martinez-Cantin, Mac Schwager, Ana C. Murillo, and Eduardo Montijano. Gen-swarms: Adapting deep generative models to swarms of drones. In Alessio Del Bue, Cristian Canton, Jordi Pont-Tuset, and Tatiana Tommasi, editors, *Computer Vision – ECCV 2024 Workshops*, pages 85–101, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-91813-1.

[103] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, volume 1, pages 399–406. IEEE, 1995.

[104] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, 2009.

[105] Sean Quinlan and Oussama Khatib. Elastic bands: Connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 802–807. IEEE, 1993. doi: 10.1109/ROBOT.1993.291936.

[106] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103:126–133, 6 1981. ISSN 0022-0434.

[107] Shunan Ren, Ying Xie, Xiangdong Yang, Jing Xu, Guolei Wang, and Ken Chen. A method for optimizing the base position of mobile painting manipulators. *IEEE Transactions on Automation Science and Engineering*, 14(1):370–375, 2017. doi: 10.1109/TASE.2016.2612694.

[108] Open Robotics. Gazebo sim: An open-source robotics simulator, 2022. Version Ignition Fortress.

[109] Loris Roveda, Jeyhoon Maskani, Paolo Franceschi, Arash Abdi, Francesco Braghin, Lorenzo Molinari Tosatti, and Nicola Pedrocchi. Model-based reinforcement learning variable impedance control for human-robot collaboration. *Journal of Intelligent & Robotic Systems*, 100(2):417–433, 2020.

[110] Taku Senoo, Gaku Jinnai, Kenichi Murakami, and Masatoshi Ishikawa. Deformation control of a multijoint manipulator based on Maxwell and Voigt models. *IEEE International Conference on Intelligent Robots and Systems*, pages 2711–2716, 2016.

[111] Taku Senoo, Kenichi Murakami, and Masatoshi Ishikawa. Deformation Control of a Manipulator Based on the Zener Model. *Journal of Robotics and Mechatronics*, 31(2): 263–273, 2019.

[112] J Shao, G Xie, and L Wang. Leader-following formation control of multiple mobile vehicles. *IET Control Theory & Applications*, 1(2):545–552, 2007.

[113] Abdel-Nasser Sharkawy, Panagiotis N Koustoumpardis, and Nikos Aspragathos. A neural network-based approach for variable admittance control in human–robot cooperation: online adjustment of the virtual inertia. *Intelligent Service Robotics*, 13(4): 495–519, 2020.

[114] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer International Publishing, 2 edition, 2016. ISBN 978-3-319-32550-7.

[115] Doganay Sirintuna, Alberto Giammarino, and Arash Ajoudani. An object deformation-agnostic framework for human–robot collaborative transportation. *IEEE Transactions on Automation Science and Engineering*, 21(2):1986–1999, 2024. doi: 10.1109/TASE.2023.3259162.

[116] PAL Robotics S.L. Tiago pro, 2025. URL https://docs.pal-robotics.com/25.01/tiagopro.html.

[117] Siciliano B Slotine and Bruno Siciliano. A general framework for managing multiple tasks in highly redundant robotic systems. In *proceeding of 5th International Conference on Advanced Robotics*, volume 2, pages 1211–1216, 1991.

[118] Chiara Landi Talignani, Federica Ferraguti, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Admittance control parameter adaptation for physical human-robot interaction. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[119] Francesco Tassi and Arash Ajoudani. Multi-modal and adaptive robot control through hierarchical quadratic programming. *Journal of Intelligent & Robotic Systems*, 110: 164, 11 2024. ISSN 1573-0409.

[120] Francesco Tassi, Elena de Momi, and Arash Ajoudani. Augmented hierarchical quadratic programming for adaptive compliance robot control. *Proceedings - IEEE International Conference on Robotics and Automation*, 2021-May:3568–3574, 2021. ISSN 10504729.

[121] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005. ISBN 9780262201629. doi: 10.1017/S0269888906210993.

[122] Elio Tuci, Muhanad H. M. Alkilabi, and Otar Akanyeti. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers in Robotics and AI*, Volume 5 - 2018, 2018. ISSN 2296-9144. doi: 10.3389/frobt.2018.00059.

[123] Thijs Ha Van den Broek, Nathan van de Wouw, and Henk Nijmeijer. Formation control of unicycle mobile robots: a virtual structure approach. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8328–8333. IEEE, 2009.

[124] E Velenis, P Tsiotras, and C Canudas-de Wit. Extension of the lugre dynamic tire friction model to 2d motion. In *Proceedings of the 10th IEEE Mediterranean conference on control and automation-MED*, pages 9–12, 2002.

[125] Hsiu-Ming Wu and Muhammad Qomaruz Zaman. Enhanced hierarchical fuzzy formation control with fuzzy collision avoidance behavior for multiple mecanum wheeled mobile robots. *Robotics and Autonomous Systems*, 194:105124, 2025. ISSN 0921-8890.

[126] Aizhen Xie, Teng Chen, Xuewen Rong, Guoteng Zhang, Yibin Li, and Yong Fan. A robust and compliant framework for legged mobile manipulators using virtual model control and whole-body control. *Robotics and Autonomous Systems*, 164:104411, 2023. ISSN 0921-8890.

[127] Ruihan Yang, Yejin Kim, Rose Hendrix, Aniruddha Kembhavi, Xiaolong Wang, and Kiana Ehsani. Harmonic mobile manipulation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3658–3665. IEEE, 2024.

[128] Xinbo Yu, Wei He, Qing Li, Yanan Li, and Bin Li. Human-robot co-carrying using visual and force sensing. *IEEE Transactions on Industrial Electronics*, 68(9):8657–8666, 2020.

[129] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236. Ieee, 2007.

[130] Novak Zagradjanin, Dragan Pamucar, and Kosta Jovanovic. Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method. *Symmetry*, 11(10):1241, 2019.

[131] Weijian Zhang, Charlie Street, and Masoumeh Mansouri. A decoupled solution to heterogeneous multi-formation planning and coordination for object transportation. *Robotics and Autonomous Systems*, 180:104773, 2024. ISSN 0921-8890.

[132] Yixiao Zhang, Qi Wang, Yue Shen, Ning Dai, and Bo He. Multi-auv cooperative control and autonomous obstacle avoidance study. *Ocean Engineering*, 304:117634, 2024. ISSN 0029-8018.