University of Udine

DRIM

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

Administrative Headquarters: University of Genoa

Trajectory Planning and Optimization for Human-Robot Collaboration

by

Federico Lozer

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

January 2026

Prof. Alessandro Gasparetto	Supervisor
Prof. Antonio Sgorbissa	Head of the PhD program

Thesis Jury:

Prof. Lorenzo Scalera, *University of Udine*	Internal examiner
Prof. Paolo Gallina, *University of Trieste*	External examiner
Dr. Nick Van Oosterwyck, *University of Antwerp*	External examiner

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
Università di Genova

Borsa di dottorato cofinanziata con risorse dell'Unione europea-*NextGeneration EU*
Piano Nazionale di Ripresa e Resilienza Missione 4, componente 1 "*Potenziamento dell'oferta dei servizi di istruzione: dagli asili nido all'Università*

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.

Federico Lozer
January 2026

Abstract

Collaborative robotics is reshaping the landscape of automation by enabling safe, efficient, and intuitive interaction between humans and robots in shared workspaces. Unlike traditional industrial robots that operate in isolation, cobots are designed to work alongside humans, requiring advanced capabilities in perception, decision-making, and motion planning to ensure seamless cooperation. Trajectory planning in collaborative robotics involves generating feasible paths that respect kinematic constraints, avoid collisions, and adapt to real-time changes in the workspace.

This thesis investigates safety strategies for shared workspaces, emphasizing speed and separation monitoring to prevent collisions while maintaining workflow fluency. Several robot stopping strategies are experimentally compared using a 7-degrees-of-freedom (DOFs) Franka Emika Panda robot. The best-performing approach dynamically scales safety zones to improve collaboration fluency and is successfully validated in an industrial case study, automating the precise placement of small components on car rear lamps, meeting both safety and cycle time requirements.

Minimum-jerk trajectory planning for generating smooth and predictable robot movements are also investigated. A mixed time–jerk optimization framework is implemented and tested on the same robotic platform, ensuring reduced acceleration and jerk while maintaining task efficiency. An extended method for redundant manipulators further optimizes both timing and joint configurations, minimizing end-effector jerk.

Moreover, this thesis proposes a neural network-based trajectory planner that enables the robot to mimic human arm movements. Trained on a custom dataset, the network selects kinematic configurations that yield human-like motion patterns, improving trust, predictability, and cooperation between humans and robots. Simulations and experimental validations demonstrate the effectiveness of this approach in achieving intuitive, human-like robotic behavior.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Trajectory planning and optimization are fundamental processes in robotics, autonomous vehicles, aerospace systems, and other dynamic applications where motion must be carefully controlled. Trajectory planning involves generating a feasible path or sequence of states that a system should follow to move from an initial position to a desired goal while satisfying kinematics and dynamics constraints, while avoiding environmental obstacles. Trajectory optimization, on the other hand, refines this path to achieve optimal performance according to specific criteria, such as minimizing time, energy consumption, or control effort.

Modern approaches to trajectory planning and optimization often combine mathematical modeling, control theory, and numerical optimization techniques. These techniques range from classical methods like gradient descent and dynamic programming to modern approaches leveraging reinforcement learning, evolutionary algorithms, and model predictive control.

Applications of collaborative trajectory planning span diverse sectors, including advanced manufacturing, healthcare (e.g., surgical assistance and rehabilitation), logistics, and service robotics. As cobots become more intelligent and context-aware, trajectory planning and optimization will play a centered role in achieving fluid, human-centric collaboration that enhances productivity, safety, and user experience.

This thesis explores the trajectory planning related to the collision avoidance, for industrial purposes. Additionally, it investigates trajectory optimization in the context of both time and jerk reduction with classical methods and human-like motion with an innovative neural-network-based technique.

Part I of this thesis talks about trajectory planning for human-robot collaboration, putting the attention on safety strategies for the coexistance in the workspace. It explores and experimentally compares the effectiveness of robot stopping approaches based on the speed

and separation monitoring for improving fluency in collaborative robotics. The safety approach is implemented to stop the robot in the event of a potential collision with the human operator. In the compared approaches, a supervisory controller checks the distance between the bounding volumes enclosing human operator and robot, and prevents potential collisions by determining the robot stop time and triggering a stop trajectory if necessary. The methods are tested on a Franka Emika robot with 7-DOFs, involving 27 volunteer participants, who are asked to walk along assigned paths to cyclically intrude the robot workspace, while the manipulator is working. The best-performing method is also tested on a real use-case: the automation of the sticking of small components on car rear lamps, a swift and repetitive operation that is currently performed manually by an operator on a factory assembly line. The challenging automation of this process includes the delicate pick and place of tiny items in predefined positions on the rear lamps. Pick-and-place trajectories for the robot are planned and a custom end-effector is designed to accomplish the task. The experimental results show that scaling online the dynamic safety zones is beneficial for improving fluency of human-robot collaboration, showing significant statistical differ- ences with respect to alternative approaches. The feasibility of the proposed approach in the assembly line, while meeting the robot constraints and the cycle time requirements, is a plus which demonstrates its usability in a real factory scenario.

Part II is about minimum-jerk trajectory planning, with the focus on the optimization procedure. In collaborative robotics, where robots and human operators share the same workspace, it is essential that robot movements are smooth and predictable. "Smooth" trajectories, continuous motions without sudden accelerations or abrupt changes in direction, reduce the risk of accidents and increase the human's sense of safety. Harmonious movements allow the operator to anticipate the robot's actions, facilitating cooperation and improving workflow efficiency. Moreover, well-planned trajectories reduce mechanical stress on the robot's components, contributing to its longevity and reliability. Optimal time-jerk trajectories for robotic manipulators are investigated in this section. An experimental setup has been developed to validate the execution of smooth motion profiles passing through a sequence of via-points, obtained by optimizing a mixed time–jerk cost function. The tests are carried out on a seven-degree-of-freedom Franka Emika robot, equipped with accelerometers to capture motion-induced oscillations at the end-effector. The results of the experiments are consistent with the numerical simulations and confirm the effectiveness of the proposed approach for generating smooth and feasible trajectories for robotic manipulators. Furthermore, an innovative minimum-jerk trajectory planning method for redundant manipulators is introduced. The approach could be seen as an evolution of the previous one: not only

optimizes the time intervals between consecutive waypoints along the assigned path, but also determines the optimal configuration of a selected robot joint in order to minimize the jerk of the end-effector. This multi-stage optimization framework is validated through extensive numerical simulations as well as experimental trials on a seven-degree-of-freedom robot performing a pick-and-place task. Experimental evidence, supported by accelerometer measurements of end-effector vibrations, demonstrates that the proposed method effectively reduces both acceleration and jerk levels of the redundant manipulator when compared with a state-of-the-art trajectory planning technique.

Part III investigates human-like trajectory planning with the help of a neural network. The objective of this work proposed in this chapter is to design a trajectory planner capable of replicating human arm movements on a redundant collaborative robot, specifically the Franka Emika Panda robot. The proposed application plans and optimizes trajectories over predefined waypoints, selecting kinematic configurations that allow the manipulator to reach a target pose in a manner resembling human motion. By ensuring that the robot adopts configurations similar to those of the human arm, human-robot collaboration can be enhanced in shared workspaces. This similarity increases trust and predictability, as the operator can more easily interpret and anticipate the manipulator's movements. The proposed method employs a neural network trained on a custom-built dataset to select, among infinite possible solutions, the one that best reflects human arm configurations. The application was experimentally tested in simulation and validated by comparing the execution of tasks performed by a human subject.

All the previous activities were carried on in the Mechatronics and Robotics Laboratory (mec-rob-lab.uniud.it) and the AI4HRC (Artificial Intelligence For Human-Robot Collaboration) laboratory of the University of Udine, the second project of the Part I has been done in collaboration with Marelli Automotive Lighting s.p.a during my 8 months traineeship period in factory, in Tolmezzo (UD), and the work described in Part III was the project I accomplished during my traineeship period abroad at ADMiRE Research center in Villach, Austria.

Part I

Trajectory planning for human-robot collaboration

Chapter 2

Introduction to trajectory planning

In the context of Industry 4.0, collaborative robotics represents one of the most promising technological paradigms, where robots are not isolated behind safety cages but are instead directly integrated into modern manufacturing processes. This integration allows safe human-robot interaction, enabling workers to operate alongside robotic systems in the same workspace without the need for physical barriers. Such interaction is possible thanks to advances in sensing, control, and safety frameworks that guarantee risk minimization. The benefits are manifold: increased efficiency, higher productivity, and improved flexibility in production, all while ensuring worker safety and wellbeing Hanna et al. (2022); Li et al. (2023a). In practice, collaborative robots (cobots) are not simply tools for automation but are designed to complement human operators, exploiting the precision and endurance of machines together with the adaptability and problem-solving skills of humans. As a result, they have become central to the concept of "smart factories" that lie at the core of Industry 4.0.

The integration of collaborative robotics in industry has particularly revolutionized assembly tasks, where human dexterity and decision-making combine with the accuracy and repeatability of robots. This synergy allows the execution of operations that would otherwise be too complex for full automation, while also relieving humans from highly repetitive or ergonomically demanding activities. Several concrete examples of such collaborative systems are described in the literature, with a large concentration in the automotive industry. For instance, in Andronas et al. (2023) the implementation of a human-high payload robot symbiotic workstation is validated through a case study involving the assembly of body parts on a vehicle chassis, a traditionally heavy and repetitive task. Similarly, in Gašpar et al. (2020) a reconfigurable robot work cell is proposed for low-volume production, showing its applicability across five industrial processes from different sectors, including the assembly

of automotive light housings. The work in Nemec et al. (2022) introduces a flexible and reconfigurable work cell for the assembly of car starters, which not only enables adaptability to different product variants but also leverages technologies for the online recognition of human intentions and the real-time learning of robust assembly policies. These studies illustrate how collaborative robotics can bring flexibility, adaptability, and learning capabilities to traditionally rigid production environments.

To achieve such close human-robot cooperation, ensuring safety becomes a fundamental requirement. Specific measures must be introduced to minimize the risk of injuries during accidental contacts Huang et al. (2022); Quiñones et al. (2023); Seriani et al. (2018), or to prevent unsafe interactions altogether through advanced collision avoidance strategies Merckaert et al. (2022); Wang et al. (2023). Over the years, a variety of approaches have been developed to address this challenge. Many of these rely on online trajectory scaling and motion re-planning, dynamically modifying the robot's path or velocity to avoid collisions in real time. These strategies are often framed within the speed and separation monitoring (SSM) paradigm, defined in ISO/TS 15066 Standard (2016), which prescribes the safe control of collaborative robots by continuously monitoring the separation distance between human and robot and adapting the robot's behavior accordingly.

The implementation of SSM can be found in several works. For example, in Lacevic et al. (2022) the concept of danger zones is introduced, explicitly modeling robot surroundings to avoid unintended collisions. The approach in Pereira et al. (2022) proposes a method to ensure non-collision while maximizing robot up-time, thus addressing not only safety but also production efficiency. A computationally efficient planning and control architecture that integrates a Rapidly-exploring Random Tree (RRT) planner with a trajectory-based Explicit Reference Governor is described in Merckaert et al. (2024), where human skeletons are approximated with spheres and cylinders to simplify calculations. Real-time motion control for manipulators operating in dynamic environments is presented in Merckaert et al. (2022), showing that even in unpredictable human motion scenarios, risk-free coexistence can be achieved. Other approaches include trajectory scaling based on safety evaluation for mobile manipulators Lippi and Marino (2020), and dynamic SSM using scene semantics obtained from thermal and depth cameras Yang et al. (2022). Collectively, these studies demonstrate the evolution from static safety concepts to adaptive, perception-driven frameworks for guaranteeing safe collaboration.

Beyond SSM-based strategies, motion planning approaches play a critical role in enabling safe and efficient collaboration. In Faroni et al. (2022), the inclusion of a human model in the robot's path planner is used to minimize execution times while handling pauses or slowdowns

caused by human presence. Haptic feedback devices combined with human-aware motion planning are explored in Grushko et al. (2021) to improve mutual understanding between human and robot, creating a more transparent collaboration. Adaptive robot motion to balance smoothness, execution time, and safety constraints is described in Lagomarsino et al. (2022), highlighting the trade-offs inherent in collaborative workspaces. A robust collision avoidance method using repulsive force generation and trajectory optimization is proposed in Liu et al. (2022), whereas Braglia et al. (2023) integrates B-splines and hidden Markov models for safe online planning validated on a Franka Emika arm. The most recent works, such as Liu et al. (2024a), rely on advanced control frameworks that combine model predictive control with stochastic control barrier functions, ensuring safety in dynamic, human-involved environments. These developments highlight the progressive sophistication of motion planning, moving from reactive to predictive and optimization-driven frameworks.

In addition to trajectory adaptation, many safety frameworks combine the SSM paradigm with power and force limiting (PFL). PFL assumes that a collision cannot always be avoided but ensures that the energy transferred to the human remains below injury thresholds. A representative study is provided in Svarny et al. (2019), where both SSM and PFL are applied in collaborative tasks with a KUKA LBR iiwa robot. Similarly, Kim et al. (2021) explores safety in assembly tasks by accounting for runaway motion and physical interaction, ultimately balancing safety with operational efficiency.

Another widely explored strategy consists in the geometric simplification of robots and humans into safety representations. These models allow efficient collision detection and safety zone computation in real time. For instance, in Byner et al. (2019) manipulator parts are approximated as spheres, enabling robot speed modulation depending on operator proximity. Capsules are used in Safeea et al. (2019); Secil and Ozkan (2022) to model both robots and humans, facilitating on-the-fly path adjustments. Control barrier functions for safety-related controllers are studied in Ferraguti et al. (2022), while voxel-based elastic bands are introduced in Kot et al. (2022). The work in Li et al. (2023b) proposes distance fields that leverage robot kinematics to generalize signed distance functions to arbitrary configurations, offering a more flexible representation. Reduced models based on 3D offsets of links are employed in Choi et al. (2022) for mixed reality frameworks, whereas Lacevic et al. (2022) defines danger zones via triangular meshes or convex approximations. Intention and trajectory prediction for collision-free collaboration using capsule models is presented in Lyu et al. (2022), while Pereira et al. (2022) compares dynamic and static safety zones, also evaluating the human co-worker's perception of safety.

Finally, dynamic safety zone approaches have been developed as an evolution of the SSM paradigm. In Scalera et al. (2020), sphere swept lines (SSLs) are employed to encapsulate human and robot bodies, enabling efficient real-time computation of distances and online collision checks. Unlike simpler models, SSLs provide a continuous representation of moving links, improving both accuracy and efficiency. This method is extended in Scalera et al. (2022, 2021), where the optimization of safety zones considers robot dynamics constraints, allowing stop times to be minimized online without compromising safety. These contributions illustrate a trend toward increasingly sophisticated safety mechanisms that merge control theory, perception, and geometric modeling.

In summary, collaborative robotics within Industry 4.0 is enabled not only by mechanical design and automation but also by a wide range of safety, planning, and control strategies. From reconfigurable work cells and intention recognition to dynamic SSM and predictive motion planning, the literature demonstrates that safe human-robot collaboration is achievable through the interplay of multiple methodologies. The continuous development of these techniques points to a future in which humans and robots can seamlessly and safely share workspaces, unlocking unprecedented levels of flexibility and efficiency in manufacturing.

Differently from previous works, this section explores and compares experimentally the effectiveness of different robot stopping strategies based on dynamic safety zones with multiple subjects, considering human-robot collaboration fluency metrics. The methods are extensively tested on a 7-degree-of-freedom robot arm, involving 27 volunteer participants, who are asked to walk along assigned paths to periodically enter the robot workspace while the manipulator is working. The experimental results show that scaling oThe adoption of dynamic safety zones has shown clear benefits for enhancing fluency in collaborative robotics. In particular, significant statistical differences are reported when compared to alternative approaches, such as the linear search of the best stop time Scalera et al. (2020) or the implementation of ISO/TS 15066 with static safety zones Standard (2016). With respect to static safety zones, improvements of 60.7% and 53.0% in total task time are obtained by the optimal scaling of safety zones and by the linear search of the stop time, respectively. These results confirm that dynamic safety zones represent a more effective and adaptive solution for safe human-robot collaboration.

In parallel, a practical case study is presented on the automation of sticking small components onto car rear lamps, an operation that is currently carried out manually by operators. The Tests performed on a UR5e manipulator demonstrate the feasibility of the operation while ensuring compliance with the robot's physical limitations and meeting the cycle time requirements of the task.

The main scientific contributions of this section can be outlined as follows:

- the experimental comparison of different robot stopping strategies based on dynamic safety zones for planning safe stop trajectories in case of possible human-robot collisions;
- the results of an extensive experimental campaign with multiple volunteer participants, assessing the performance of the compared strategies through real-time interaction with the robot during bespoke experimental sessions;
- the use of multiple human-robot collaboration fluency metrics as quantitative measures for statistical analysis, evaluating both task completion time and additional indicators that capture the overall quality of the human-robot collaboration;
- the implementation of the proposed safety approach in a real industrial case study, together with an evaluation of the effect of cycle time on quantitative collaboration metrics and on the robot's mechanical energy consumption.

Chapter 3

An experimental evaluation of robot stopping approaches for improving fluency in collaborative robotics

3.1 Problem Statement

We compare three different collision avoidance strategies for collaborative robotics, all based on bounding volumes that enclose both the human and the robot to ensure safety and improve fluency. The strategies experimentally evaluated in this chapter follow the *Speed and Separation Monitoring* (SSM) paradigm, since it is one of the most advanced collaborative modalities described in ISO/TS 15066 that do not involve physical contact between the operator and the robot during motion.

In all the approaches considered, a supervisory controller continuously verifies compliance with the SSM requirements. Whenever a potential unintended collision is detected, the controller commands the robot to stop without violating the manipulator kinematic or dynamic limits. According to ISO/TS 15066 Standard (2016), the robot must never approach the human closer than the *protective separation distance* S_p, which is computed online as:

$$S_p = S_h + S_r + S_s + \xi \tag{3.1}$$

where:

- S_h is the distance traveled by the human, moving at the maximum speed $v_h = 1.6$ m/s specified in ISO/TS 15066, during the reaction time t_r of the supervisory controller and the robot stop time t_s;

- S_r is the distance traveled by the robot, moving at its maximum speed v_r during the reaction time t_r;
- S_s is the distance traveled by the robot during its stop time t_s;
- ξ accounts for undetected human body parts, sensing system tolerances, and robot localization uncertainties.

To represent both human and robot, the strategies compared in this chapter use *sphere swept lines* (SSLs). An SSL for a generic robot link is defined as the Minkowski sum of a sphere of radius r_v, the smallest radius required to encompass the link geometry, and a line segment with endpoints a_k and b_k (Fig. 3.1).

Based on this representation, *safety zones* are defined as SSLs expanded to include the protective separation distance S_p, thereby overestimating the distance required for the robot to stop from its current kinematic state. For a generic link, the safety zone radius is computed as:

$$r_{sz} = r_v + S_p \tag{3.2}$$

Finally, a potential collision is detected when the distance between any pair of human–robot line segments becomes smaller than the sum of their respective bounding radii. In such cases, the stop trajectory is immediately activated.

3.2 Proposed Approach

In this chapter, we analyze three collision avoidance strategies, all characterized by different methods for calculating the robot stop time:

- **Approach 1:** optimal scaling of dynamic safety zones Scalera et al. (2022, 2021).
- **Approach 2:** scaling of dynamic safety zones with linear search of the stop time Scalera et al. (2020).
- **Approach 3:** static safety zones, corresponding to a basic implementation of ISO/TS 15066 Standard (2016).

All three approaches comply with the ISO/TS 15066 specification, with no distinction made between operators with or without mobility issues or prior experience. However, they affect collaboration fluency differently, which directly influences the productivity and cycle

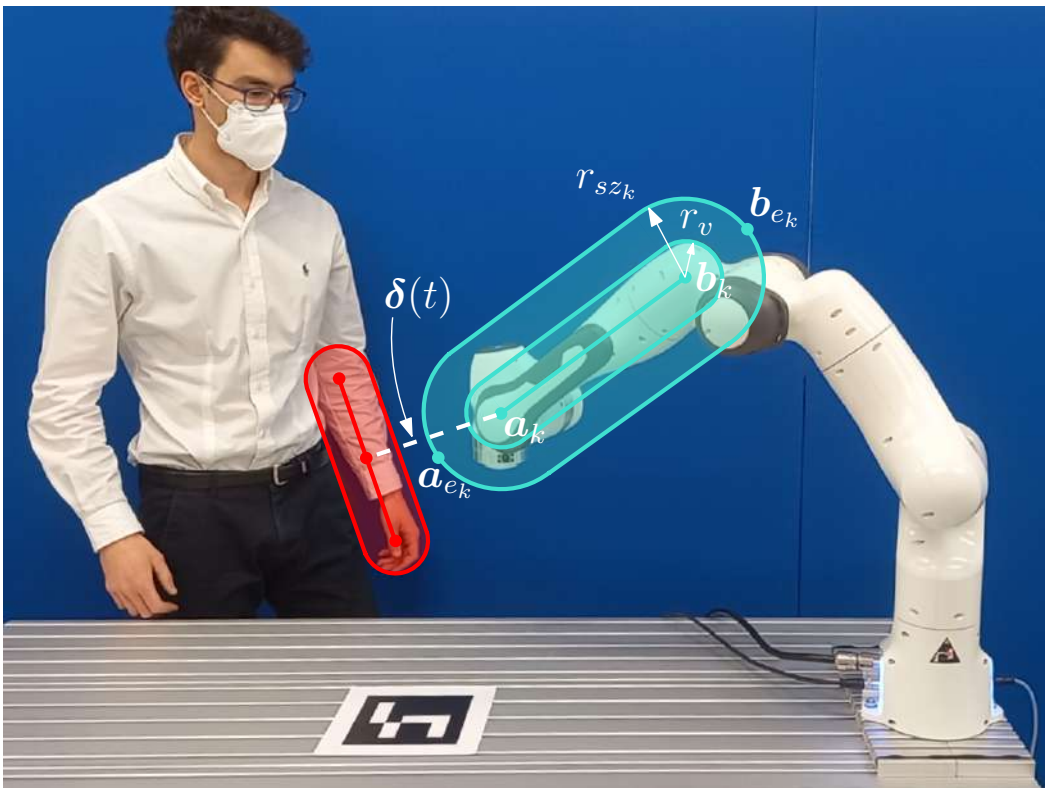


Figure 3.1 A SSL and the corresponding safety zone encapsulate the k^{th} part of the manipulator (light blue), and a bounding volume encloses one body part (red). The minimum-distance line segment is illustrated as dashed line (white).

time of the robotic system Hoffman (2019); Scalera et al. (2022). A brief description of each approach follows.

3.2.1 Approach 1: Optimal Scaling of Dynamic Safety Zones

Approach 1 relies on the online scaling of dynamic safety zones Scalera et al. (2022, 2021). The main idea is to minimize online the stop time t_s required by the robot to halt safely from its current kinematic state. Once t_s is defined, a candidate stop trajectory q_s is planned as a 5^{th}-order polynomial in the robot joint space, from the kinematic state at $t_0 + t_r$ (initial joint position $q_{s,i}$, velocity $\dot{q}_{s,i}$, and acceleration $\ddot{q}_{s,i}$) to the final state at $t_0 + t_r + t_s$, defined by $q_{s,f} = q_{s,i}$, $\dot{q}_{s,f} = 0$, and $\ddot{q}_{s,f} = 0$, where t_0 is the current time.

The optimization problem that determines the stop time t_s is:

$$\min_{t_s} \quad w_0 t_s + w_1 \left| t_s - t_{s,\text{prev}} \right| \tag{3.3}$$

subject to

$$
\begin{aligned}
q_{i,\min} \leq q_{s,i} \leq q_{i,\max} \\
|\dot{q}_{s,i}| \leq \dot{q}_{i,\max} \\
|\ddot{q}_{s,i}| \leq \ddot{q}_{i,\max} \\
|\dddot{q}_{s,i}| \leq \dddot{q}_{i,\max} \\
|\tau_i(q_s, \dot{q}_s, \ddot{q}_s)| \leq \tau_{i,\max} \\
|\dot{\tau}_i(q_s, \dot{q}_s, \ddot{q}_s)| \leq \dot{\tau}_{i,\max} \\
t \in [t_0 + t_r, t_0 + t_r + t_s], \quad i = 1, \ldots, N
\end{aligned} \tag{3.4}
$$

Here, w_0 and w_1 are positive weights, while $t_{s,\text{prev}}$ is the stop time computed in the previous iteration. Including $t_{s,\text{prev}}$ ensures smooth variations in safety zone radii between iterations. The variables $q_{i,\min}$ and $q_{i,\max}$ denote the joint position limits, while $\dot{q}_{i,\max}$, $\ddot{q}_{i,\max}$, $\dddot{q}_{i,\max}$, $\tau_{i,\max}$, and $\dot{\tau}_{i,\max}$ are the respective limits of velocity, acceleration, jerk, torque, and torque rate at joint i. Feasibility requires respecting all kinematic and dynamic limits. Finally, N is the number of manipulator DOFs.

To verify torque constraints, the joint torques τ along the stop trajectory q_s are computed as:

$$
\tau = M(q_s)\ddot{q}_s + C(q_s, \dot{q}_s)\dot{q}_s + F_v\dot{q}_s + f_c \operatorname{sign}(\dot{q}_s) + g(q_s) \tag{3.5}
$$

where $M(q_s)$ is the mass matrix, $C(q_s, \dot{q}_s)$ represents Coriolis and centrifugal terms, while viscous friction, Coulomb friction, and gravity are modeled by $F_v\dot{q}_s$, $f_c \operatorname{sign}(\dot{q}_s)$, and $g(q_s)$, respectively. The torque rate $\dot{\tau}$ is computed as the incremental ratio of torque over time. In this work, the robot dynamics are assumed to be perfectly known, although interval arithmetic can be used to ensure torque limit compliance under model uncertainty Giusti and Nainer (2022); Scalera et al. (2023).

Once t_s is obtained, the safety separation distance S_p is computed. The contributions used in each approach are summarized in Table 3.1. Specifically, S_h assumes the human moves at the maximum ISO/TS 15066 speed v_h, S_r is computed from the maximum speed v_{s_k} of each robot part during the reaction time, and S_s is the time integral of the maximum link velocity component v_{δ_k} along the shortest human–robot distance direction:

$$
v_{\delta_k}(t) = \max\left(\left|\dot{a}_{e_k}(t) \cdot \delta(t)\right|, \ \left|\dot{b}_{e_k}(t) \cdot \delta(t)\right|\right) \tag{3.6}
$$

where $\dot{a}_{e_k}$ and $\dot{b}_{e_k}$ are the velocities of the spherical end-caps encapsulating link k, and $\delta(t)$ is the minimum-distance direction between human and manipulator, as described in Scalera et al. (2022, 2021).

3.2.2 Approach 2: Scaling of Dynamic Safety Zones with Linear Search

Approach 2 differs from Approach 1 because it avoids online optimization. The stop trajectory is still modeled by a 5^{th}-order polynomial, but the stop time t_s is selected from a predefined set of candidate values $t_s = [t_{s,1}, \ldots, t_{s,n}]$ with $t_{s,j} < t_{s,\text{j+1}}$ and $t_{s,n} \leq t_{s,\text{wc}}$ (the worst-case stop time). The chosen t_s is the first candidate that satisfies the manipulator's kinematic and dynamic constraints.

As shown in Table 3.1, this approach does not project link velocities onto the shortest human–robot distance vector in computing S_s. While simpler, it does not guarantee smooth safety zone radii across iterations and does not fully exploit robot dynamics during a safety stop. However, its computational efficiency makes it useful when online optimization (Eq. (3.3)–(7.2)) cannot be reliably solved within the reaction time Scalera et al. (2020).

3.2.3 Approach 3: Static Safety Zones

Approach 3 implements the most basic SSM strategy defined by ISO/TS 15066 Standard (2016), resulting in static safety zones. The stop time is fixed at the worst-case value $t_{s,\text{wc}}$, leading to constant safety zone radii that ignore the current robot kinematics and dynamics.

Table 3.1 Contributions of S_p in each approach.

	S_h	S_r	S_s
Approach 1	$v_h\,(t_r + t_s)$	$v_{s_k}\,t_r$	$\int_{t_0+t_r}^{t_0+t_r+t_s} v_{\delta_k}(t)\,dt$
Approach 2	$v_h\,(t_r + t_s)$	$v_r\,t_r$	$\int_{t_0+t_r}^{t_0+t_r+t_s} v_{s_k}(t)\,dt$
Approach 3	$v_h\,(t_r + t_{s,\text{wc}})$	$v_r\,t_r$	$v_r\,t_{s,\text{wc}}$

3.3 Experimental Setup

The experiments are performed with a Franka Emika Panda arm featuring 7-DOFs (Figure 3.2). The robot is operated through the Robot Operating System (ROS) Melodic Morenia, running on a workstation with Ubuntu 18.04 LTS Bionic Beaver, an Intel Core

i5-10600k processor, 32 GB RAM, and an Intel UHD Graphics 630 GPU. The control is implemented in Python.

The interface with the robot controller is managed by *libfranka*, the C++ client-side implementation of the Franka Control Interface. This library enables execution of real-time commands, reading of robot status at 1 kHz, and access to model libraries for computing kinematic and dynamic terms. The supervisory controller runs at 20 Hz (limited by the human-pose tracking system described in Subsection 3.3.1). ROS integration is achieved using *franka_ros*, with *MoveIt* employed for path planning, and the Robotics Toolbox for Python Corke and Haviland (2021) for kinematic description via rigid body tree representation. The joint position controller is adopted to steer the manipulator during the experimental trials.

The optimization of the stop time t_s, subject to the kinematic and dynamic constraints defined in Eq. (3.3) with (7.2), is solved online using the *IPOPT* algorithm from the nonlinear optimization framework *CasADi* Andersson et al. (2019). The symbolic robot dynamics are provided via the *urdf2casadi* library Johannessen et al. (2019), based on the URDF model of the manipulator. Model parameters are taken from the experimentally validated dynamic model in Gaz et al. (2019). The joint space limits of the Franka Emika Panda are reported in Table 3.2.

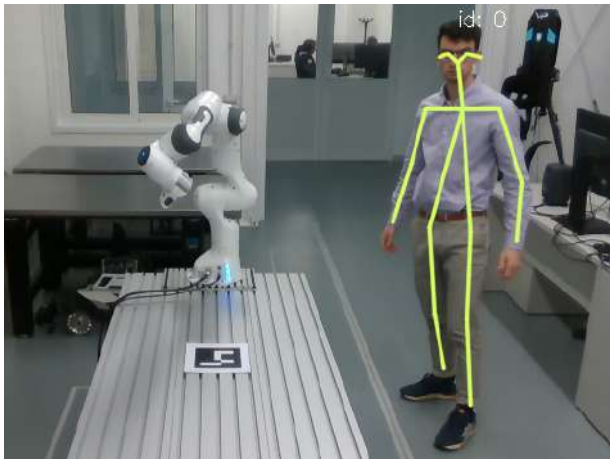
This setup with the Franka Emika arm is representative of a typical collaborative robotics scenario: the manipulator is lightweight, provides joint torque sensing, and its payload-to-weight ratio is consistent with other robots of similar size Haddadin et al. (2022).

Table 3.2 Joint space limits of the Franka Emika Panda GmbH (2017).

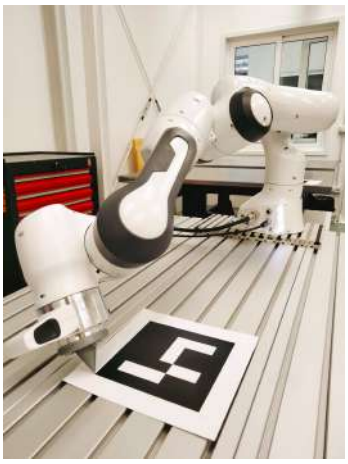
Limit		**Joint**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
$q_{\min}$	[rad]	−2.8973	−1.7628	−2.8973	−3.0718	−2.8973	−0.0175	−2.8973
$q_{\max}$	[rad]	2.8973	1.7628	2.8973	−0.0698	2.8973	3.7525	2.8973
$\dot{q}$	[rad/s]	2.1750	2.1750	2.1750	2.1750	2.6100	2.6100	2.6100
$\ddot{q}$	[rad/s^2]	15	7.5	10	12.5	15	20	20
$\dddot{q}$	[rad/s^3]	7500	3750	5000	6250	7500	10000	10000
τ	[Nm]	87	87	87	87	12	12	12
$\dot{\tau}$	[Nm/s]	1000	1000	1000	1000	1000	1000	1000

3.3.1 Perception system

An Intel RealSense D435 depth camera is employed to track the human operator during the experiments. Skeleton tracking is performed online using the *CubeMOS Skeleton Tracking*



(a) Camera frame with skeleton tracking of a participant.



(b) Robot equipped with calibration tool and ArUco marker.

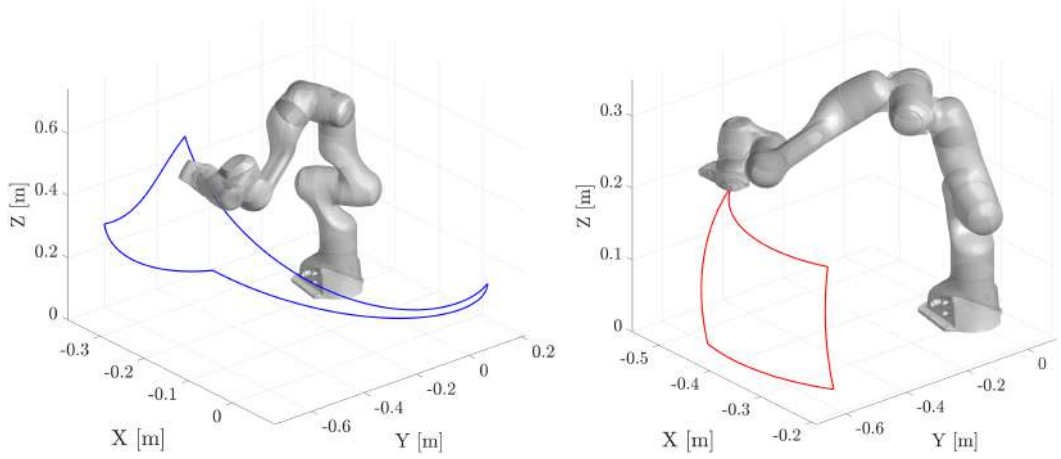
Figure 3.2 Experimental setup.

SDK. Eighteen body joints are identified, but only those required to reconstruct the arms, head, and chest are retained. The human is thus modeled with six bounding volumes, sized according to anthropometric data of British adults Pheasant and Haslegrave (2018).

Initially, body joints are extracted in pixel coordinates and combined into a stylized 2D skeleton (Figure 3.2a). The algorithm converts these into 3D Cartesian coordinates in the camera frame. To evaluate distances between robot and human, the joints must be transformed into the robot base frame.

This transformation is obtained using an *ArUco* marker placed near the robot base. The camera-to-marker transformation is estimated with the C++ *ArUco* library Garrido-Jurado et al. (2016); Romero-Ramirez et al. (2018), relying on *OpenCV* for marker-based 3D pose estimation. The marker-to-robot transformation is determined with a 3D printed calibration tool attached to the end-effector, by manually touching the marker corners (Figure 3.2b). Combining these two transformations yields the camera-to-robot transformation matrix, enabling conversion of human joint coordinates into the robot base frame.

Finally, the minimum distance between human and robot is computed online between pairs of line segments representing the bounding volumes of the manipulator and human. The shortest-distance algorithm used Ericson (2004) is efficient enough for real-time applications.



(a) Test A. (b) Test B.

Figure 3.3 Desired paths for the robot.

Table 3.3 Specifications of the point-to-point motions for Test A (Figure 3.3a).

Time [s]	q_1 [rad]	q_2 [rad]	q_3 [rad]	q_4 [rad]	q_5 [rad]	q_6 [rad]	q_7 [rad]
0	-1.846	0.632	-0.290	-1.626	0.126	2.268	-0.036
2.5	-1.553	0.689	0.497	-1.473	0.566	2.746	0.128
5.0	-0.485	0.763	0.846	-2.340	0.512	2.565	0.287
7.5	-0.974	0.0174	-0.509	-1.851	-1.591	1.019	0.226

3.3.2 Test Protocol

The experimental tests aim to evaluate and compare the collision avoidance strategies described in Section 6.1, within a scenario of human-robot collaboration tested by multiple human subjects.

The collaborative task consists of the robot arm performing a point-to-point motion among four randomly selected joint-space positions of the manipulator. Nominal trajectories are planned with 5^{th} order polynomials with zero initial and final velocity and acceleration. The specific target points are reported in Tables 3.3 and 3.4 for the experiments shown in Figures 3.3a and 3.3b, respectively. The robot requires a nominal time of 10 s to reach each point and return to the initial position. This cycle is repeated three times, yielding a total of 30 s (excluding additional safety stops and restarts).

During robot execution, the human operator walks along a predefined path marked on the laboratory floor (Figure 3.4) and performs cyclical intrusions into the robot workspace to

Table 3.4 Specifications of the point-to-point motions for Test B (Figure 3.3b).

Time [s]	q_1 [rad]	q_2 [rad]	q_3 [rad]	q_4 [rad]	q_5 [rad]	q_6 [rad]	q_7 [rad]
0	-1.266	0.937	0.027	-1.565	-0.033	2.405	-0.427
2.5	-1.318	0.376	0.042	-1.608	0.003	1.978	1.222
5.0	-1.955	0.256	-0.388	-1.961	0.090	2.273	-0.240
7.5	-1.853	1.092	-0.067	-1.367	0.109	2.477	-1.894

trigger safety stops. The human trajectory is not strictly controlled: each subject naturally varies in walking style, body dimensions, and motion patterns, leading the robot to stop and restart at different time instants. Participants are instructed to maintain a constant walking speed until the robot completes its task. The supervisory controller does not adapt to instantaneous variations in human speed; instead, the human motion space S_h is conservatively estimated assuming the worst-case maximum walking speed ($v_h = 1.6$ m/s), in accordance with ISO/TS 15066.

Each participant completed two tests:

- **Test A:** the human walks back and forth along the linear blue path in Figure 3.4a, while the robot executes the trajectory in Figure 3.3a. Each subject repeats Test A three times, once per each of Approaches 1, 2, and 3.

- **Test B:** the human walks counterclockwise around the robot table along the red path in Figure 3.4b, while the robot executes the trajectory in Figure 3.3b. Each subject repeats Test B twice, once per each of Approaches 1 and 2.

Test A allows a comparison of all three approaches, whereas Test B highlights the differences between Approaches 1 and 2. Approaches 1 and 2 employ dynamic safety zones, which enable the human to move closer to the robot during motion. In contrast, Approach 3 relies on large static safety zones that prevent robot movement when the human is near its workspace (e.g., walking by the table).

A total of 27 individuals participated in the study (age range: 22–40 years, mean: 27.3 years, standard deviation: 4.8 years). Eight participants had prior experience with human-robot collaboration. None of the subjects reported mobility issues.

Before beginning the tests, participants provided informed consent and were instructed to maintain a constant walking speed along the designated path, without modifying their behavior in response to robot movements. Each participant performed five experiments in total (three for Test A, two for Test B), presented in random order. Subjects were not informed of which collision avoidance approach was active in each trial.

3.4 Experimental Results

Experimental data were processed in *Matlab* to extract quantitative metrics for comparing the approaches. Robot positions and velocities were sampled at 1 kHz, while stop times, safety zone radii, human-robot distances, and human joint positions were recorded at the supervisory controller rate.

The first level of comparison examines robot stop time and safety zone radii, which directly indicate the effectiveness of each strategy. Subsequently, we evaluate the practical impact on collaboration using quantitative fluency metrics, which assess the quality of shared-location teamwork between human and robot. Fluency metrics were first introduced in Hoffman and Breazeal (2007) for anticipatory controllers in shared workspaces, and later extended in Dragan et al. (2015); Hoffman (2019).

In this work, we consider four metrics:

- *Total task time* (T-TIME): the overall time required by the robot to complete its task, including interruptions due to safety stops. This is directly linked to system productivity.

- *Robot idle time* (R-IDLE): the percentage of T-TIME during which the robot is stationary or awaiting restart. Idle time is identified when the joint velocity norm is below 0.05 m/s. Lower values of R-IDLE correspond to more fluent and productive collaboration.

- *Concurrent activity in the robot workspace* (C-ACT-WS): a variant of the standard concurrent activity metric Hoffman (2019), representing the percentage of T-TIME in which both human and robot are simultaneously active *within the robot workspace*. This measures collaborative fluency in close contact and the reactivity of the robot Scalera et al. (2022). In these tests, the human is considered active throughout while inside the workspace.

- *Number of robot stops* (R-STOPS): the number of safety stops triggered during task execution, i.e., the number of times a stop trajectory is activated.

Subjective evaluations were not considered in order to focus on objective, quantitative comparisons. Statistical analysis was carried out as follows: the Shapiro-Wilk (S-W) test Shapiro and Wilk (1965) was used to verify normality of continuous variables. For normally distributed outcomes, statistical differences between two stopping approaches were assessed with the paired-sample t-test Student (1908). For non-normal distributions, the Kruskal-Wallis (K-W) H-test Kruskal and Wallis (1952) was applied.

Figures 3.5 and 3.6 summarize the experimental results for Test A and Test B, respectively. The box plots display the distributions of robot stop time, radii of the safety zones, T-TIME, R-IDLE, C-ACT-WS, and R-STOPS across participants for the compared stopping approaches. The corresponding quantitative metrics (mean, standard deviation, and percentage difference between pairs of approaches) are reported in Tables 3.5 and 3.6.

Stop Time

For Test A, the stop times are normally distributed for Approach 1 (S-W test $p = 0.3537$), but not for Approaches 2 and 3 ($p = 0.0058$ and $p < 0.001$, respectively). Statistical analysis (K-W test) reveals that Approach 1 achieves significantly lower stop times than both Approaches 2 and 3 ($p < 0.001$ in all cases), with improvements of 15.0% (1–2), 57.5% (1–3), and 50.0% (2–3).

In Test B, stop times are normally distributed for both approaches (S-W test $p = 0.3067$ and $p = 0.1661$ for Approaches 1 and 2, respectively). A paired-sample t-test shows a significant difference ($p < 0.001$), with Approach 1 providing faster stops of the robot (9.5% improvement).

Safety Zone Radii

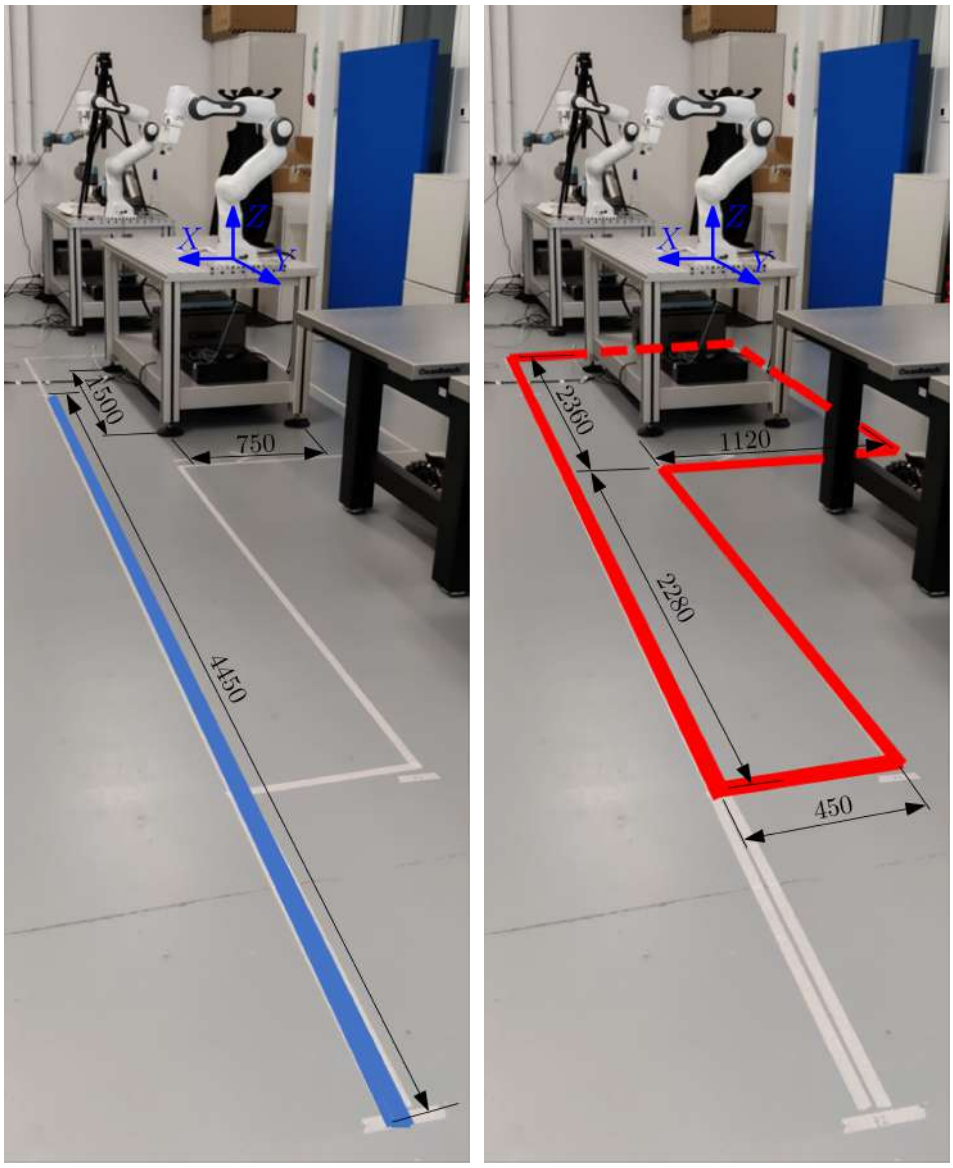
In Test A, the radii of the robot safety zones present different normality behaviors:

- Approach 1: all radii are non-normally distributed ($p = 0.0253$, $p = 0.0278$, $p = 0.0191$, $p = 0.0279$ for r_{sz_1}–r_{sz_4}).
- Approach 2: only r_{sz_1} and r_{sz_2} are normally distributed ($p = 0.0613$, $p = 0.0632$), while r_{sz_3} and r_{sz_4} are not ($p = 0.0373$, $p = 0.0274$).
- Approach 3: all radii are non-normally distributed ($p < 0.001$ for all).

Pairwise comparisons show that the radii are significantly smaller for Approach 1 compared to Approach 2 (K-S test $p = 0.0013$, $p = 0.0013$, $p < 0.001$, $p < 0.001$), for Approach 1 compared to Approach 3 ($p < 0.001$ for all), and for Approach 2 compared to Approach 3 ($p < 0.001$, $p < 0.001$, $p = 0.0053$, $p = 0.005$).

In Test B, safety zone radii are non-normally distributed for Approach 1 (S-W test $p = 0.0171$, $p = 0.0111$, $p = 0.0054$, $p = 0.0037$), but normally distributed for Approach 2 (S-W test $p = 0.9991$, $p = 0.9943$, $p = 0.8682$, $p = 0.7998$). A significant difference is again found between the two approaches (K-W test $p < 0.001$ for all radii), with Approach 1 outperforming Approach 2.

These trends are evident in the box plots of Figures 3.5 and 3.6, as well as in the percentage differences summarized in Tables 3.5 and 3.6.



(a) Test A. (b) Test B.

Figure 3.4 Paths for the human operator. Measures are in millimeters.

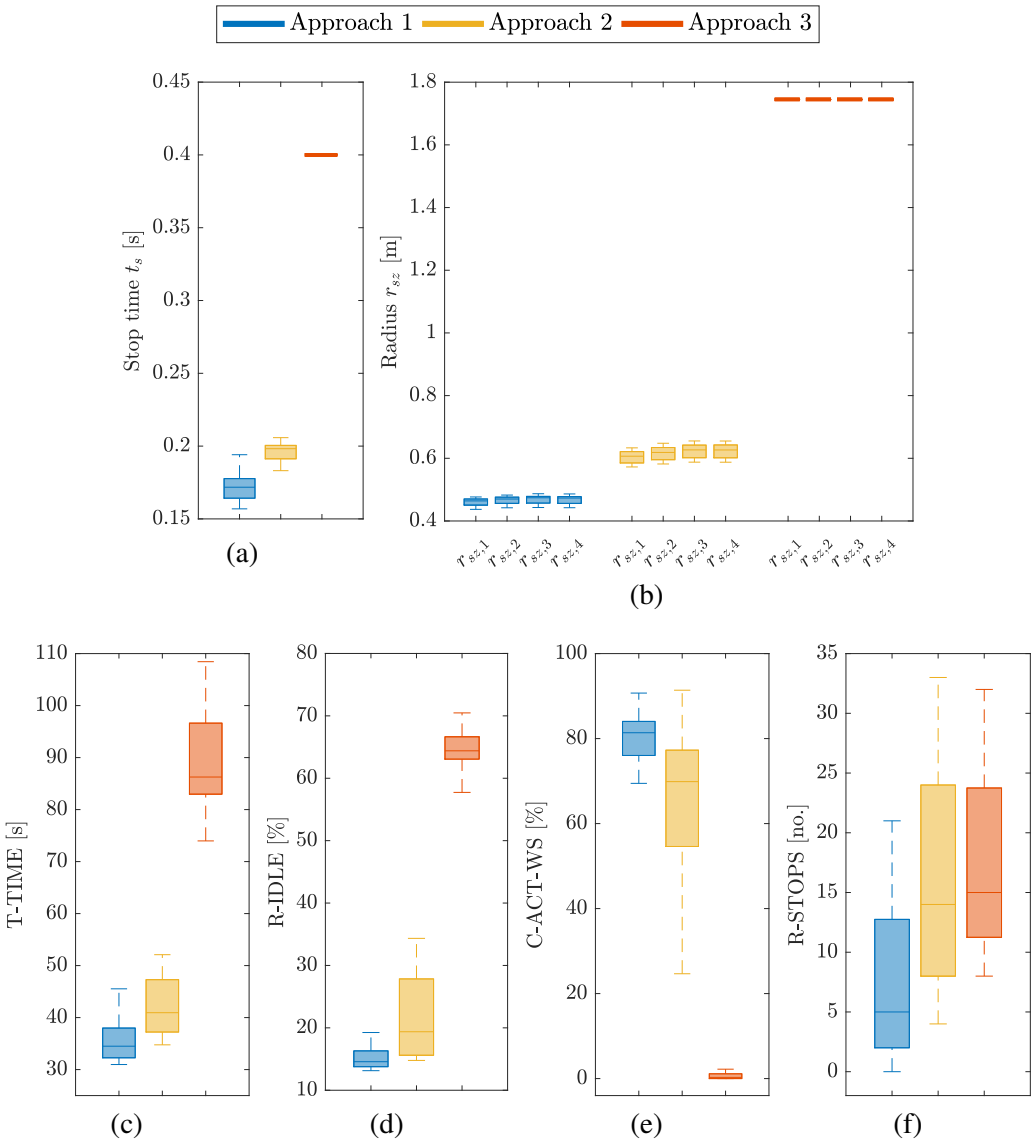


Figure 3.5 Experimental results for Test A: stop time (a), radii of the safety zones (b), T-TIME (c), R-IDLE (d), C-ACT-WS (e), and R-STOPS (f).

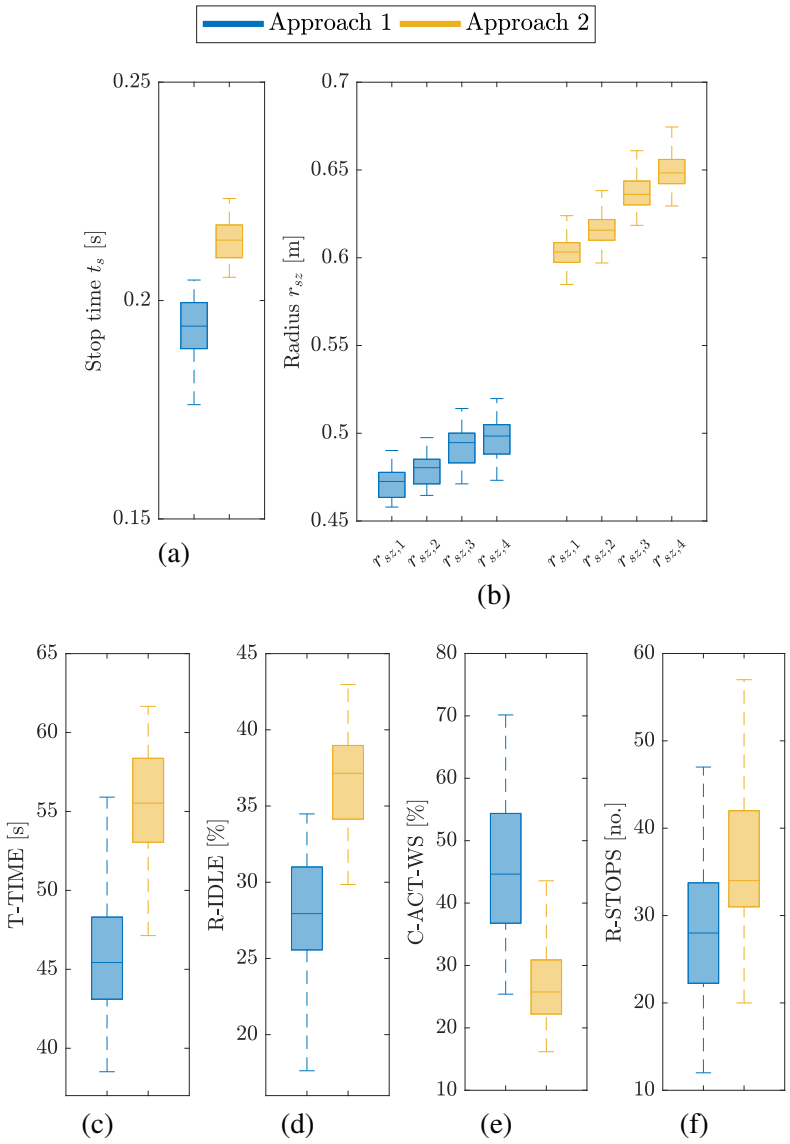


Figure 3.6 Experimental results for Test B: stop time (a), radii of the safety zones (b), T-TIME (c), R-IDLE (d), C-ACT-WS (e), and R-STOPS (f).

Fluency Metrics

The values of stop times for Test A are normally distributed for Approach 1 (S-W test $p = 0.3537$), but not for Approaches 2 and 3 ($p = 0.0058$ and $p < 0.001$, respectively). Stop times of Approach 1 result significant lower than Approaches 2 and 3 (K-W test: 1-2 $p < 0.001$, 1-3 $p < 0.001$, and 2-3 $p < 0.001$), with improvements of 15.0% (1-2), 57.5% (1-3), and 50.0% (2-3). Considering Test B, stop times are found to be normally distributed (S-W test $p = 0.3067$, $p = 0.1661$ for Approach 1 and 2, respectively). A significant difference is

Table 3.5 Mean ± standard deviation for stop time, radii of the safety zones, and fluency metrics resulted from Test A.

$\Delta\%$ indicates the percentage difference between the means of pairs of approaches.
○: not significative; ◐: p-value < 0.05; ●: p-value < 0.01.

	Approach			**$\Delta\%$**			***p*-value**		
Metric	**1**	**2**	**3**	**(1-3)**	**(2-3)**	**(1-2)**	**(1-3)**	**(2-3)**	**(1-2)**
t_s [s]	0.17 ± 0.01	0.20 ± 0.01	0.40	57.5 %	50.0 %	15.0 %	●	●	●
$r_{sz,1}$ [m]	0.46 ± 0.01	0.60 ± 0.02	1.75	73.7 %	65.7 %	23.3 %	●	●	●
$r_{sz,2}$ [m]	0.47 ± 0.01	0.62 ± 0.02	1.75	73.1 %	64.6 %	25.0 %	●	●	●
$r_{sz,3}$ [m]	0.47 ± 0.01	0.62 ± 0.02	1.75	73.1 %	64.6 %	25.0 %	●	●	●
$r_{sz,4}$ [m]	0.47 ± 0.01	0.62 ± 0.02	1.75	73.1 %	64.6 %	25.0 %	●	●	●
T-TIME [s]	35.5 ± 3.9	42.5 ± 5.7	90.4 ± 11.5	60.7 %	53.0 %	16.5 %	●	●	◐
R-IDLE [%]	15.8 ± 3.4	21.8 ± 6.6	65.0 ± 3.5	75.9 %	66.7 %	27.5 %	●	●	◐
C-ACT-WS [%]	79.8 ± 10.4	65.2 ± 15.8	0.6 ± 0.7	99.2 %	81.0 %	18.3 %	●	●	◐
R-STOPS [*no.*]	7.3 ± 6.4	16.3 ± 8.9	20.1 ± 13.8	63.7 %	18.9 %	55.2 %	●	○	●

found between Approach 1 and 2 in Test B (paired-sample t-test $p < 0.001$), where Approach 1 performs better with faster stops of the robot (9.5%).

In Test A, the radii of the robot safety zones are non-normally distributed for Approach 1 ($p = 0.0253$, $p = 0.0278$, $p = 0.0191$, $p = 0.0279$ for r_{sz_1}, r_{sz_2}, r_{sz_3}, and r_{sz_4}); only r_{sz_1} and r_{sz_2} are normally distributed for Approach 2 ($p = 0.0613$, $p = 0.0632$, $p = 0.0373$, $p = 0.0274$); all radii are non-normally distributed for Approach 3 ($p < 0.001$ for all safety zones). The radii of the safety zones are smaller with Approach 1 with respect to 2 (K-S test $p = 0.0013$, $p = 0.0013$, $p < 0.001$, $p < 0.001$); between 1 and 3 ($p < 0.001$ for all radii), and between 2 and 3 ($p < 0.001$, $p < 0.001$, $p = 0.0053$, $p = 0.005$). Considering Test B, radii of the safety zones are non-normally distributed for Approach 1 (S-W test $p = 0.0171$, $p = 0.0111$, $p = 0.0054$, $p = 0.0037$), but are normally distributed for Approach 2 (S-W test $p = 0.9991$, $p = 0.9943$, $p = 0.8682$, $p = 0.7998$). A significant difference is found also in Test B, where Approach 1 outperforms Approach 2 also in this case (K-W test $p < 0.001$ for all the four radii). These trends can be also noticed from the box plots of Figures 3.5 and 3.6, and the percentages reported in Tables 3.5 and 3.6.

By taking into account fluency metrics for collaborative robotics, the values of T-TIME for Test A exhibit a non-normal distribution (S-W test $p = 0.0396$, $p = 0.0106$, $p = 0.0112$ for Approaches 1, 2 and 3, respectively). Lower total task times are found for Approach 1, followed by Approach 2 (K-W test: 1-2 $p = 0.0149$, 1-3 $p < 0.001$, 2-3 $p < 0.001$). In particular, with Approach 1 the task is concluded in the 60.7% of less time with respect to Approach 3, and in the 53.0% of less time with respect to Approach 2. Furthermore, the

Table 3.6 Mean ± standard deviation for stop time, radii of the safety zones, and fluency metrics resulted from Test B.

Δ% indicates the percentage difference between the means of pairs of approaches.
○: not significative; ◐: p-value < 0.05; ●: p-value < 0.01.

Metric	**Approach 1**	**Approach 2**	**Δ% (1-2)**	***p*-value (1-2)**
t_s [s]	0.19 ± 0.01	0.21 ± 0.01	9.5 %	●
$r_{sz,1}$ [m]	0.47 ± 0.01	0.60 ± 0.01	21.7 %	●
$r_{sz,2}$ [m]	0.48 ± 0.01	0.62 ± 0.01	22.6 %	●
$r_{sz,3}$ [m]	0.49 ± 0.02	0.64 ± 0.01	23.4 %	●
$r_{sz,4}$ [m]	0.49 ± 0.02	0.65 ± 0.01	24.6 %	●
T-TIME [s]	45.8 ± 3.9	55.4 ± 4.2	17.3 %	●
R-IDLE [%]	28.1 ± 3.8	36.8 ± 3.6	23.6 %	●
C-ACT-WS [%]	45.6 ± 10.6	27.1 ± 7.6	40.6 %	●
R-STOPS [*no.*]	28.3 ± 8.4	35.6 ± 9.2	20.5 %	●

robot completes the operation 16.5% faster with Approach 2 than Approach 3. Better results in terms of T-TIME are found for Approach 1 also in Test B, where values are normally distributed across participants (S-W test $p = 0.6199$, $p = 0.1289$ for Approach 1 and 2, respectively). The task was completed 17.3% faster by the robot with Approach 1 than Approach 2 in Test B (paired-sample t-test $p < 0.001$).

A trend similar to T-TIME is found for R-IDLE. In Test A, values of robot idle time are non-normally distributed across participants for Approaches 1 and 2 (S-W test $p < 0.001$ and $p = 0.0026$), whereas present a normal distribution for Approach 3 (S-W test $p = 0.6467$). The percentage of the T-TIME that the robot has remained still during the execution of the task is significant lower for Approach 1, followed by Approach 2 (K-W test: 1-2 $p = 0.0183$, 1-3 $p < 0.001$, 2-3 $p < 0.001$), with percentage improvements of 75.9% (1-3), 66.7% (2-3), and 27.5% (1-2). In Test B, values of robot idle time are normally distributed across subjects (S-W test $p = 0.4022$, $p = 0.3963$ for Approach 1 and 2, respectively). In Test B, R-IDLE is found to be 23.6% lower for Approach 1 than with Approach 2 (paired-sample t-test $p < 0.001$).

The values of C-ACT-WS are not distributed normally for Approach 1 and 3 (S-W test $p < 0.001$ and $p = 0.0011$), whereas follow a normal distribution for the second considered robot stopping approach (S-W test $p = 0.1711$). The Kruskal-Wallis test reports significant difference between Approaches 1 and 2 ($p = 0.0165$), 1 and 3 ($p < 0.001$), 2 and 3 ($p < 0.001$). From Figure 3.5 and Table 3.5, it can be noted that almost no concurrent activity in the robot workspace is possible with Approach 3. Furthermore, Approach 1 performs 18.3% better than Approach 2. C-ACT-WS is significant higher (40.6%) for Approach 1

than Approach 2 also in Test B (paired-sample t-test $p < 0.001$), where values are normally distributed (S-W test $p = 0.7694$, $p = 0.1748$ for Approach 1 and 2, respectively).

Finally, the numbers of robot stops are non-normally distributed across participants in Test A ($p = 0.0193$, $p = 0.0423$, and $p < 0.001$ for Approach 1, 2 and 3, respectively). In this case, Approach 1 performs better than 2 and 3 (K-W test: 1-2 $p = 0.001$, 1-3 $p < 0.001$) with percentage improvements of 63.7% (1-3) and 55.2% (1-2), but no coherent difference in the number of robot stops is found between Approaches 2 and 3 ($p = 0.6216$). The values of R-STOPS are normally distributed in Test B (S-W test $p = 0.9239$, $p = 0.3080$ for Approach 1 and 2, respectively), and a significant reduction (20.5%) of robot stops is found when using the optimal scaling of dynamic safety zones with respect to the approach based on the linear search of the stop time (paired-sample t-test $p < 0.001$).

Overall, the use of dynamic safety zones and optimized robot stop times provides statistically significant improvements not only in robot stop time and safety zone radii, but also across all fluency metrics (T-TIME, R-IDLE, C-ACT-WS, and R-STOPS). In particular, scaling robot safety zones online enables closer human-robot interaction (as evidenced by C-ACT-WS), resulting in more efficient collaboration with fewer safety stops. Conversely, the static safety zone approach is effective for safety but limits human proximity during robot motion, leading to poor fluency. The linear search of stop time serves as a compromise when online optimization tools are unavailable, though with notably reduced performance compared to optimal scaling.

3.5 Summary

This chapter explored and experimentally compared stopping approaches for collaborative robotics based on speed and separation monitoring (SSM), with the goal of improving fluency. In all approaches, a supervisory controller monitors distances between bounding volumes enclosing the human and the robot, and triggers a stop trajectory when potential collisions are detected. The methods were tested with 27 volunteer participants, who walked along predefined paths and periodically intruded into the robot workspace during operation.

The results demonstrate that scaling dynamic safety zones online leads to significant improvements in fluency metrics. In particular, reductions of 60.7% and 53.0% in T-TIME are observed with the optimal scaling method and the linear search method, respectively, compared to static safety zones.

These findings highlight that the choice of strategy depends on application requirements:

- If minimizing total task time is critical and sufficient computational resources are available, Approach 1 (optimal scaling) is preferable.
- If maximizing C-ACT-WS is the goal, only Approaches 1 and 2 are viable, since no concurrent activity is possible with static safety zones.
- If limited computational resources prevent online optimization, Approaches 2 or 3 are the only feasible solutions.

These insights become particularly relevant for heavier robots, where stop times are strongly influenced by robot dynamics.

Chapter 4

A collaborative robotics application for the assembly of car rear lamps

4.1 Problem statement

This chapter addresses the automation of a manual procedure for sticking small components onto car rear lamps with the implementation of the safety strategies discussed in Chapter 3, which prevent potential collisions between the robot and the human operator.

As a case study, we consider eight small adhesive items that must be positioned at specific locations on the back of a pair of rear lamps. Currently, the operation is performed manually by an operator who retrieves the components from a dispenser and attaches them to the desired points. This project was carried on in collaboration with Marelli Automotive Lightning s.p.a.

The workspace accommodates two rear lamps, processed within the same cycle. Each cycle requires placing four pieces per lamp (two round Goretex and two rectangular film components), for a total of eight items. The desired cycle time for the process is approximately 45 s.

The objective of this chapter is to automate the procedure with an industrial robot, thereby reducing or eliminating the need for manual intervention. The research activities include:

- Planning robot trajectories that satisfy both the cycle time and the kinematic constraints of the manipulator and the workspace.
- Investigating pick (retrieval from dispenser) and place (positioning on the lamp) operations, with attention to gripping and adhesion of the components.

- Implementing a safety strategy that halts robot motion in case of potential collision with a human operator.
- Evaluating the influence of cycle time on both quantitative human-robot collaboration metrics and the robot's mechanical energy.

4.2 Proposed Approach

Trajectories are optimized to ensure correct gripping and positioning of all four stickers within the desired cycle time, while respecting robot kinematic constraints and avoiding unwanted collisions with the worktable, rear lamps, or dispensers. The retrieval of round stickers and the placement of all four items are performed using force control, which allows the robot to detect rigid surfaces by moving at low speed and stopping upon contact. Vacuum pressure monitoring is employed to detect unsuccessful retrievals, interrupting the task if needed. The operator can choose whether to apply stickers to one or both lamps.

To ensure operator safety, the approach based on the optimal scaling of dynamic safety zones Scalera et al. (2022, 2021) is implemented. This strategy performs online safety checks between bounding volumes enclosing the robot and the human to detect potential collisions. The safety zones are defined by minimizing the robot's stop time, while considering dynamics, kinematics, torque limits, and the directed velocity of robot links relative to the human. The approach complies with the SSM criterion of ISO/TS 15066, which prescribes a protective separation distance while the robot is in motion. If this distance is violated, a stop trajectory is triggered. The optimization problem is formulated to minimize the stop time under constraints on position, velocity, acceleration, jerk, torque, and torque rate (see Scalera et al. (2022, 2021) for details).

To verify the torque constraints during the robot stop, the dynamic model identified in Boscariol et al. (2023) is employed. In this study, the dynamic model parameters are assumed to be known. However, in cases of imperfect knowledge of the robot dynamics, an interval arithmetic-based approach can be used to ensure compliance with joint torque limits Scalera et al. (2023).

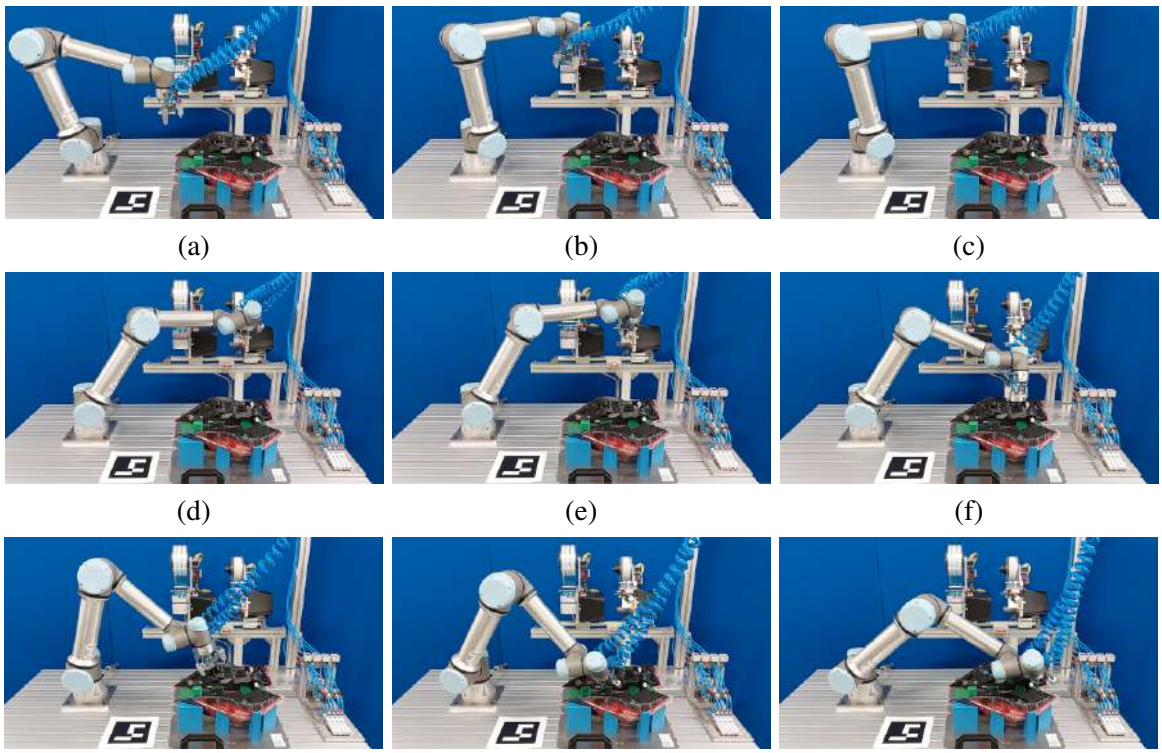


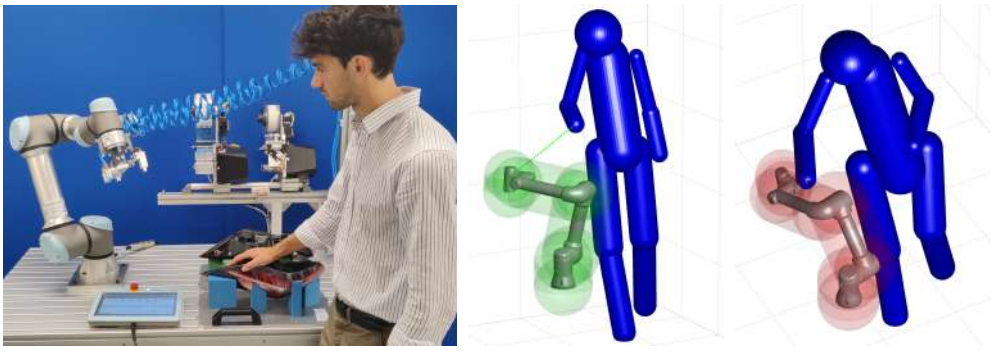
Figure 4.1 Exemplary frames of the process: (a) homing position; (b) pick first round sticker; (c) pick second round sticker; (d) pick first rectangular sticker; (e) pick second rectangular sticker; (f) place first rectangular sticker; (g) place second rectangular sticker; (h) place first round sticker; (i) place second round sticker.

4.3 Experimental Setup

The first stage of the research focuses on designing the workstation and the gripping system for the components. A UR5e manipulator by Universal Robots is adopted for this application (Fig. 4.3).

A preliminary pick-and-place program is developed in which all waypoint positions are parameterized relative to the robot base reference frame. This ensures easy adjustment of the relative positions among the rear lamps, the dispenser, and the robot base, while guaranteeing smooth robot movements free of singularities or joint limits.

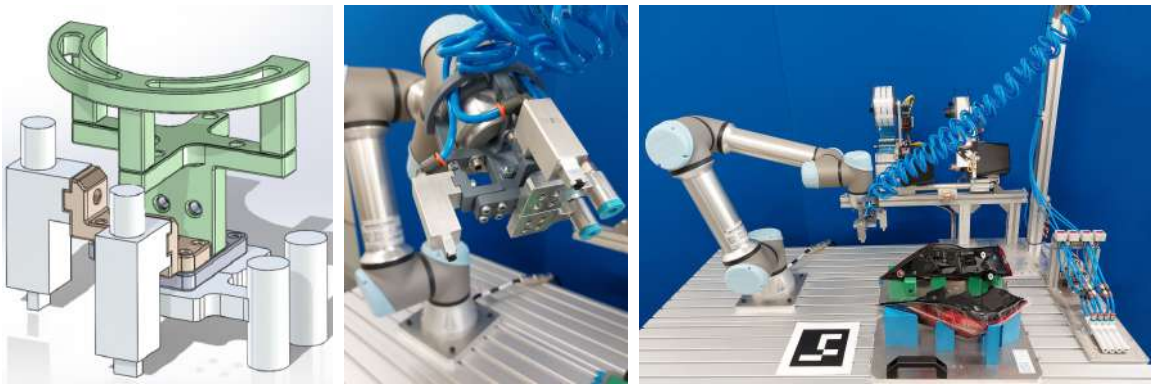
The robot end-effector is equipped with a vacuum system connected to an air compressor, designed to avoid collisions with dispensers and lamp housings (Fig. 4.3a). Four custom suction cups are mounted on ad-hoc supports (Fig. 4.3b), fabricated in PLA with an Ultimaker Pro Bundle 3D printer. With this setup, the robot retrieves components from dispensers



(a) (b) (c)

Figure 4.2 Human intrusion in the robot workspace (a); human and robot enclosed in bounding volumes keeping safety clearance (b), and during a robot stop (c).

and places them on the lamps using simple 90° rotations of its sixth joint. The design minimizes vibrations of the suction cups, prevents twisting of the air pipes, and reduces material consumption.



(a) (b) (c)

Figure 4.3 End-effector design (a) and prototype (b); robot workstation (c).

The optimization problem is implemented in Python using the open-source tool *CasADi* Andersson et al. (2019), and leverages the recursive Newton-Euler inverse dynamics of the robot in symbolic form.

The human operator's position is tracked online using an Intel RealSense D345 camera, with processing performed via CUDA and OpenPose on an NVIDIA Xavier computer.

Figure 4.4 Camera aquisitions of the experimental tests.

Once the coordinates of the human body parts are detected, as in Fig 4.4, the minimum distance between robot and human is computed efficiently by analyzing the line segments defining their respective bounding volumes. A robot stop is triggered if this distance indicates a potential collision.

4.4 Experimental Results

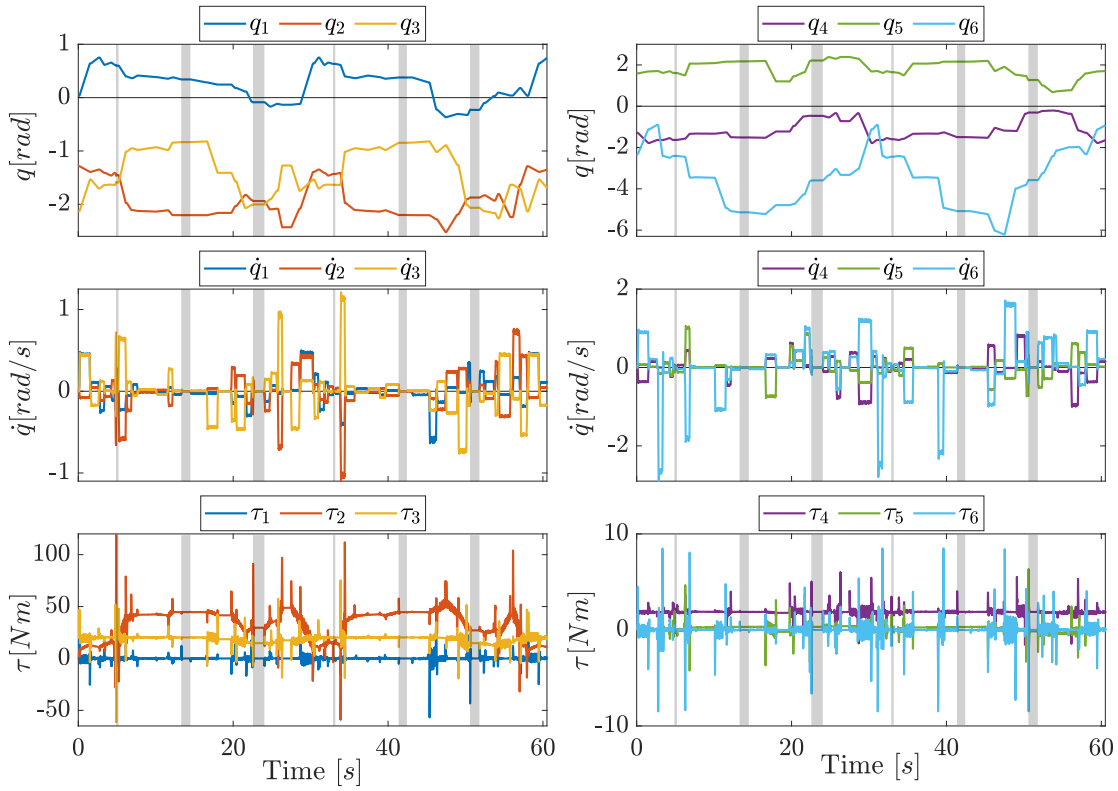


Figure 4.5 Exemplary joint positions, velocities, and torques over time for one cycle.

Figure 4.1 shows exemplary frames of the sticking process: the robot is able to complete the entire procedure of picking and placing the stickers in approximately 45 seconds, while respecting its kinematic and dynamic constraints. Fig. 4.2a illustrates a case of human intrusion in the workstation, which triggers a safety stop of the robot. Moreover, Figs. 4.2b and 4.2c present human and robot enclosed in bounding volumes, maintaining a safety clearance and during a robot stop, respectively.

To assess the impact of cycle time on quantitative metrics for human-robot collaboration, the 3D joint coordinates of the human skeleton during a workspace intrusion are recorded. These coordinates are then used in playback across multiple tests, while the robot cycle time

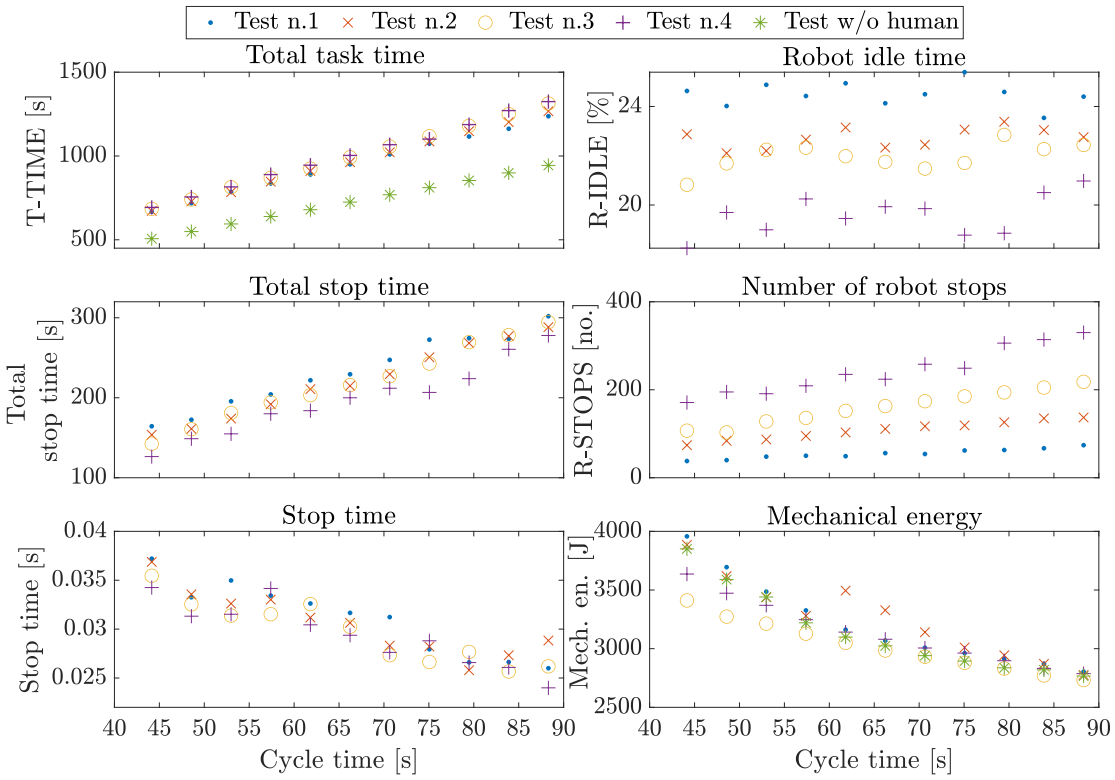


Figure 4.6 Experimental results for different values of the cycle time (ten repeated cycles are performed for each value).

is progressively increased. For each cycle time, the experiment is repeated ten times while the recorded human skeleton induces safety stops. The playback continues until the robot completes the task, i.e., assembling ten pairs of rear lamps.

The tests are further repeated four times, increasing the speed of the recorded skeleton (Test n.1: $\times 1$, Test n.2: $\times 2$, Test n.3: $\times 3$, Test n.4: $\times 4$). Fig. 4.5 shows an example trajectory for one cycle, lasting approximately 60 seconds (processing two rear lamps), including joint positions, velocities, and torques over time. Grey shaded areas indicate robot safety stops.

The following metrics are considered to evaluate the impact of cycle time on human-robot collaboration and robot energy consumption:

- Total task time (T-TIME): total time required to complete ten assemblies, including robot safety stops;
- Robot idle time (R-IDLE): percentage of T-TIME during which the robot remains stationary;
- Total stop time: cumulative duration when the robot is not moving;

- Number of robot stops (R-STOPS);
- Stop time: root-mean-square value of individual stop durations;
- Mechanical energy: integral of mechanical power over time, representing energy consumed by the robot, which is an important index to improve efficiency and sustainability Boscariol et al. (2023); Carabin and Scalera (2020).

The top-left plot in Fig. 4.6 shows that total task time increases with cycle time. Green markers indicate the nominal duration if human intrusion is ignored. No substantial differences in T-TIME are observed when varying human speed. R-IDLE (top-right plot) remains below 26% for all cycle times and decreases as skeleton speed increases. Total stop time and the number of robot stops (central plots) increase with cycle time, with more stops occurring at higher skeleton speeds. Robot stop time (bottom-left plot) decreases as cycle time increases, since the robot generally moves slower, and remains below 0.038 s. Finally, mechanical energy generally decreases with increasing cycle time, mainly due to lower robot speed. Human intrusion has negligible effect on robot energy consumption.

4.5 Summary

In this chapter, we presented the automation of sticking small items on car rear lamps. Pick-and-place trajectories for the robot were planned, and a custom end-effector was designed. A safety approach was implemented to halt the robot in case of potential collision with the human operator. Finally, the impact of cycle time on human-robot collaboration quality and robot mechanical energy was evaluated. Experimental results on a UR5e manipulator demonstrated the feasibility of the operation while meeting robot constraints and cycle time requirements.

Part II

Minimum-jerk trajectory planning

Chapter 5

Introduction to trajectory optimization

Trajectory planning has long been recognized as a fundamental research problem in robotics, and it continues to play a pivotal role in the development of modern industrial processes. The task of trajectory planning essentially consists in establishing a correlation between a predefined geometric path and a timed sequence of positions, velocities, and accelerations that a robotic manipulator must execute Biagiotti and Melchiorri (2008). Such planning is crucial because robots rarely operate in isolation: they are embedded in complex manufacturing systems where timing, synchronization, and safety are decisive. A trajectory that is carefully designed does not merely enable the robot to follow a path; it directly impacts cycle times, energy usage, component wear, and ultimately, the overall efficiency of the production line. One of the most immediate benefits is the reduction of the operational time Piazzi and Visioli (1998), which in turn translates into higher throughput, reduced costs, and improved competitiveness in production environments Chettibi et al. (2004).

Over the years, numerous approaches to trajectory planning have been investigated in the literature, each targeting different optimization criteria depending on the industrial context. Early studies emphasized time-optimal planning as a key strategy to maximize productivity. More recent contributions, however, have broadened this perspective by also addressing criteria such as energy efficiency, vibration suppression, and motion smoothness. One prominent research direction has been the optimization of robot motion with respect to energy consumption Carabin and Scalera (2020). This reflects a paradigm shift: while speed and productivity remain essential, energy consumption is increasingly seen as a strategic factor in manufacturing due to its influence on operating costs and sustainability Javaid et al. (2022). Optimized trajectories allow robots to perform the same task with less mechanical effort, avoiding unnecessary accelerations or idle movements. In practice, this not only reduces energy usage Vidussi et al. (2021); Wu et al. (2021), but also contributes to lowering

the carbon footprint of manufacturing systems, aligning robotic operations with modern sustainability goals Fabris et al. (2024b).

Another significant application domain of trajectory planning is the mitigation of motion-induced oscillations. Industrial robots, especially those with lightweight structures, are prone to unwanted vibrations that can compromise accuracy, increase mechanical stress, and shorten actuator lifetime. Ensuring smooth, jerk-limited trajectories is therefore a priority in both traditional and collaborative robotic applications Trigatti et al. (2018). The importance of smooth trajectories has grown with the rise of collaborative robotics Flowers and Wiens (2024); Seriani et al. (2018); Solak and Ajoudani (2023), where robots share the workspace with human operators. Unlike classical industrial robots, which are often rigid and operate behind safety fences, collaborative robots are deliberately designed to be more compliant and flexible to guarantee human safety. However, these very features make them more susceptible to oscillations and shocks. In this scenario, the demand for smooth, minimum-jerk trajectories is driven by both safety requirements Scalera et al. (2024a, 2021) and ergonomic considerations, since reducing vibrations can lower the cognitive workload and stress of human collaborators Lagomarsino et al. (2022). Thus, trajectory planning becomes not only a technical necessity for performance but also a human-centered design requirement.

Because smoothness, or lack thereof, is closely tied to motion-induced oscillations, jerk minimization has become a central goal in trajectory planning. A variety of strategies have been developed to achieve this. For instance, Fang et al. Fang et al. (2019) introduced an S-curve point-to-point trajectory generation method to address minimum-jerk optimization, while Dai et al. Dai et al. (2020) employed a genetic algorithm to compute minimum-jerk trajectories in the configuration space of a UR3 manipulator. Wu et al. Wu and Zhang (2022) addressed the same problem using NURBS curves, formulating it as a constrained optimization problem where jerk is reduced via a cost function defined as the sum of squared joint jerks. Earlier foundational work by Piazzi and Visioli Piazzi and Visioli (2000) approached jerk minimization through global constrained minimax optimization with cubic polynomial interpolation, while Gasparetto et al. Gasparetto and Zanotto (2007, 2008) proposed a mixed cost function balancing execution time and jerk, acknowledging the trade-off between speed and smoothness. Experimental validation by Zanotto et al. Zanotto et al. (2011) confirmed the practical benefits of this formulation, establishing it as a reference in the field. Extensive surveys such as Gasparetto et al. (2011); Ni et al. (2023) summarize the evolution of these methods and their applications.

Spline functions have also been extensively exploited in trajectory planning due to their mathematical flexibility and ability to guarantee high-order continuity. They represent an effective framework for generating smooth paths with a compact representation. One of the earliest contributions was made by Cook and Ho Cook and Ho (1984), who proposed the "434" polynomial spline method capable of ensuring acceleration continuity. This method was later refined by Boscariol et al. Boscariol et al. (2012), who introduced the "545" and "5455" spline formulations, providing continuity up to the jerk level. Such improvements in spline-based methods have proven useful in applications requiring both speed and smoothness, such as precision assembly and machining.

More recently, the research community has explored advanced optimization techniques to integrate jerk minimization with time and energy criteria. Lu et al. Lu et al. (2017a) solved the minimum time-jerk trajectory planning problem using an augmented Lagrange particle swarm optimization approach. Fang et al. Fang et al. (2019) proposed a time-optimal S-curve strategy based on piecewise sigmoid functions, generating infinitely differentiable trajectories within bounded velocity, acceleration, and jerk constraints. Building on this, Fang et al. Fang et al. (2020) proposed an improved sinusoidal jerk model for the online generation of smooth joint trajectories, with continuity up to the jerk level and short execution times. Additional examples include the adoption of quintic polynomial interpolations for minimum-time trajectories Lu et al. (2020), optimization of fifth-order B-spline curves using NSGA-II for multi-objective optimization of traveling time and jerk Huang et al. (2018), and spline-based strategies explicitly incorporating kinematic constraints Paing and Uchiyama (2022). Hybrid optimization heuristics, such as the combination of whale optimization and genetic algorithms, have also been successfully applied to trajectory planning problems with high-order smoothness requirements Lu et al. (2022b); Wu and Zhang (2022). Parallel efforts by Abu-Dakka et al. Abu-Dakka et al. (2017) focused on genetic algorithms (PPGA1/PPGA2) for minimum-time cubic spline trajectories, while Palleschi et al. Palleschi et al. (2019) developed a minimax approach for sixth-order B-spline trajectories. Collectively, these contributions highlight the trend toward sophisticated multi-objective optimization frameworks that can balance speed, smoothness, and energy efficiency in a single formulation.

It is important to note that most of the aforementioned methods are designed for traditional manipulators without redundancy, i.e., robots with a number of degrees-of-freedom exactly matching the task requirements Siciliano et al. (2009). In contrast, redundant robots, those with more DOFs than strictly necessary, offer additional flexibility that can be exploited to further improve trajectory planning performance Haug (2024); Liu et al. (2024b). For example, redundancy can be leveraged to reduce vibrations, optimize energy consumption,

or avoid kinematic singularities while maintaining task feasibility. Chang et al. Chang et al. (2021) proposed a motion planning strategy specifically aimed at vibration reduction in redundant robots. Reiter et al. Reiter et al. (2016) tackled the problem of time-optimal path following along a fixed end-effector trajectory in the case of kinematically redundant manipulators. Lu et al. Lu et al. (2017b) addressed time-jerk optimal trajectory planning for a 7-DOFs redundant robot by combining particle swarm optimization with augmented Lagrange multipliers, while Chen et al. Chen et al. (2019) developed a multilevel minimization strategy to mitigate joint-angle drift and excessive joint excursions. Finally, Dai et al. Dai et al. (2020) demonstrated the applicability of these techniques in advanced contexts such as robot-assisted 3D printing, where trajectory planning must simultaneously ensure precision, speed, and smoothness under complex motion requirements.

In summary, trajectory planning has evolved from a focus on time minimization alone to a comprehensive research field that addresses multiple performance indicators, including energy efficiency, vibration suppression, jerk reduction, and redundancy exploitation. This evolution reflects the changing priorities of robotics: from purely industrial automation focused on productivity, to modern collaborative and sustainable scenarios where efficiency, safety, and human well-being are equally critical. As robotics continues to advance, trajectory planning will remain a central research challenge, bridging mathematical optimization with real-world constraints in order to deliver high-performance, safe, and sustainable robotic operations.

In this section, we present an experimental setup to test time-jerk optimal trajectories for robotic manipulators. We implement an optimization method for the planning of smooth trajectories based on a cost function composed of a term proportional to the total execution time, and one term proportional to the squared jerk along the trajectory. The setup comprises a Franka Emika robot with 7-DOFs equipped with three accelerometers to measure the motion-induced oscillations of the end-effector. The experimental results show a good agreement with the numerical tests and demonstrate the feasibility of the approach in optimizing smooth trajectories for robotic manipulators.

We also present an improved approach for the minimum-jerk trajectory planning, specifically designed for redundant robots. The proposed approach is based on a multi-stage optimization strategy: a time-jerk optimization is first performed to reduce both the total time and the jerk of the robot end-effector, as in Lozer et al. (2023). Then, the position of one selected joint of the redundant manipulator is optimized for each of the considered waypoints of the robot path. Finally, a second time-jerk optimization is run to further improve the results of the two previous stages. The effectiveness of proposed approach is validated with

both extensive numerical simulations and experimental tests on a Franka Emika Panda arm with seven degrees-of-freedom performing a pick-and-place trajectory. The results show the feasibility of the proposed approach and the benefits in reducing the jerk levels and the measured mechanical vibrations of the robot end-effector accordingly. Furthermore, the effectiveness of the approach in improving the overall smoothness of robot trajectories is also proved with respect to a notable trajectory planning technique Gasparetto and Zanotto (2007).

To summarize, the main contributions of this section include:

- a design of an experimental setup for minimum-jerk trajectory planning of collaborative manipulators based on time intervals optimization;
- an approach for minimum-jerk trajectory planning of redundant manipulators based on a multi-stage optimization of time intervals and positions of a selected joint;
- the results of extensive numerical and experimental tests on a redundant manipulator, also including accelerometer measurements of the end-effector vibrations;
- a comparison of the proposed approach with the minimum-jerk trajectory planning technique in Gasparetto and Zanotto (2007).

Chapter 6

An experimental setup to test time-jerk optimal trajectories for robotic manipulators

6.1 Problem Statement

The planning of time-jerk optimal trajectories aims to define motion laws that are as smooth as possible while remaining fast. Limiting jerk values produces smooth movements but lengthens the trajectory duration, whereas reducing the trajectory time increases jerk. The key is to find a suitable trade-off between smoothness and speed.

6.2 Proposed Approach

The proposed algorithm parametrizes the motion profile using a set of time intervals h_i, each representing the duration of the i-th segment connecting via-point i to $i+1$. The vector $\mathbf{h}$, with $1 \leq i \leq w_p$ (number of via-points), is the decision variable vector, optimized by minimizing the cost function in Eq. (6.1), which balances the total duration T and the sum of the integral of joint jerks across N robot joints. The trade-off is set by $\alpha \in [0,1]$.

$$\min_{\mathbf{h}} \alpha \sum_{i=1}^{w_p-1} h_i + (1-\alpha) \sum_{j=1}^{N} \int_0^T (\dddot{q}_j(t))^2 \mathrm{d}t \tag{6.1}$$

The feasibility of the trajectory $(q(t), \dot{q}(t), \ddot{q}(t), \dddot{q}(t))$ is ensured by the following constraints:

$$\begin{cases} q_{\min} < q < q_{\max}, \\ |\dot{q}| < \dot{q}_{\max}, \\ |\ddot{q}| < \ddot{q}_{\max}, \\ |\dddot{q}| < \dddot{q}_{\max}, \\ |\tau| < \tau_{\max}, \\ |\dot{\tau}| < \dot{\tau}_{\max}, \end{cases} \tag{6.2}$$

where the constraints include kinematic limits and joint torques τ, computed using the inverse dynamic model:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + f_c\text{sign}(\dot{q}) + g(q) \tag{6.3}$$

Here, $M(q)$ is the mass matrix, $C(q,\dot{q})\dot{q}$ accounts for Coriolis and centrifugal effects, F_v and f_c are viscous and Coulomb friction, and $g(q)$ is gravity. The algorithm was tested with a "434" spline Boscariol et al. (2012); Cook and Ho (1984), where a third-order polynomial is used for all intermediate segments, and a fourth-order polynomial for the first and last segments. The trajectory between adjacent via-points P_k and P_{k+1} ($2 \leq k \leq N-2$) is:

$$F_k(t) = B_{k,1} + B_{k,2}t + B_{k,3}t^2 + B_{k,4}t^3 \tag{6.4}$$

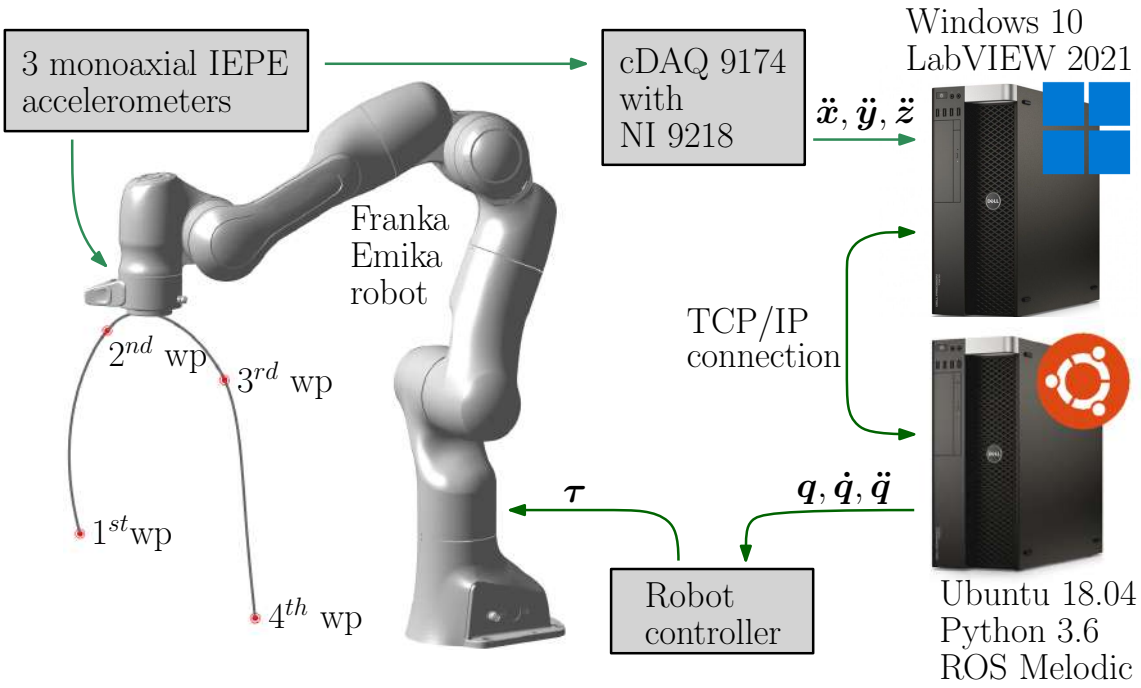
while for the first and last segments ($k = 1, N-1$):

$$F_k(t) = B_{k,1} + B_{k,2}t + B_{k,3}t^2 + B_{k,4}t^3 + B_{k,5}t^4 \tag{6.5}$$

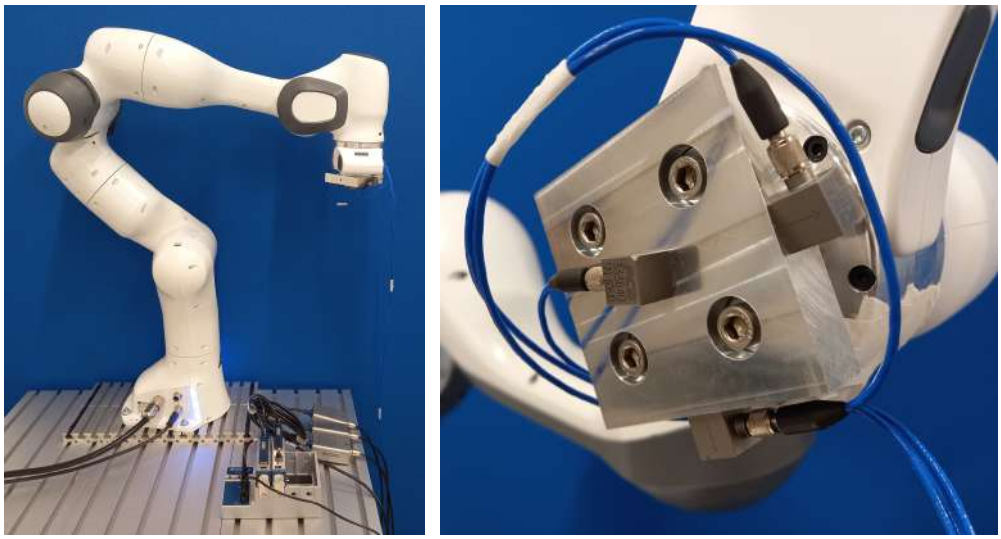
Polynomial coefficients $B_{k,i}$ are computed to ensure continuity of positions, velocities, and accelerations. Jerk continuity is not enforced, as jerk is either linearly changing or constant within each segment Cook and Ho (1984).

6.3 Experimental Setup

The optimization problem in Eqs. (6.1) and (6.2) is solved using sequential quadratic programming with the `fmincon` Matlab function. One hundred solutions were generated by varying α from 0 to 1 in equal steps. Each trajectory was executed three times on a Franka Emika 7-DOFs robot (Fig. 6.1). The end-effector is equipped with a custom aluminium flange carrying three monoaxial piezoelectric accelerometers. Acceleration data is acquired



(a)



(b) (c)

Figure 6.1 (a) Overview of the experimental setup; (b) Franka Emika robot and data acquisition system; (c) aluminium flange with accelerometers on the end-effector.

at 10 kHz using two National Instruments NI9218 C Series modules with a cDAQ 9174 chassis and filtered with a 5-sample moving-average in Matlab.

Experiments are conducted using two computers (Windows 10 and Ubuntu 18.04), which exchange data via TCP/IP. The Windows workstation generates trajectories, acquires data with LabVIEW 2021, using the block scheme in Figure 6.2, and performs post-processing in Matlab. The Ubuntu workstation controls the robot via ROS Melodic and Python 3.6, sending joint position, velocity, and acceleration sequences sampled at 100 Hz to the robot controller.

6.4 Experimental Results

Table 6.1 Results of Test 1 for some values of α.

α	T [s]	J [rad^2/s^5]	$\ddot{x}_{\text{rms}}$ [m/s^2]	$\ddot{y}_{\text{rms}}$ [m/s^2]	$\ddot{z}_{\text{rms}}$ [m/s^2]	$\ddot{X}_{\text{rms}}$ [m/s^2]
0.20	12.07	0.60	0.23	0.28	0.18	0.40
0.40	10.25	1.37	0.27	0.32	0.20	0.46
0.60	8.95	2.69	0.31	0.36	0.21	0.52
0.80	7.60	6.09	0.34	0.43	0.25	0.56
1.00	2.17	$3.51 \cdot 10^3$	1.14	1.76	1.19	2.40

Figure 6.3 presents an example of a trajectory evaluated for $\alpha = 0.5$, showing the joint positions, velocities, and absolute acceleration over time. The overall acceleration is computed as the vector sum of signals from each mono-axial accelerometer. Vertical solid lines in the figure indicate the passage through the via-points.

Measured data obtained for different values of α are summarized in Tab. 6.1. In particular, the total time T, sum of the integral of squared joint jerks J, root-mean-square Cartesian accelerations ($\ddot{x}_{\text{rms}}$, $\ddot{y}_{\text{rms}}$, $\ddot{z}_{\text{rms}}$), and the absolute root-mean-square acceleration $\ddot{X}_{\text{rms}}$ are reported. The results clearly indicate that the weight α allows the user to tune the trade-off between total execution time and end-effector acceleration. Lower α values emphasize jerk minimization, while higher α prioritizes faster trajectories with increased accelerations.

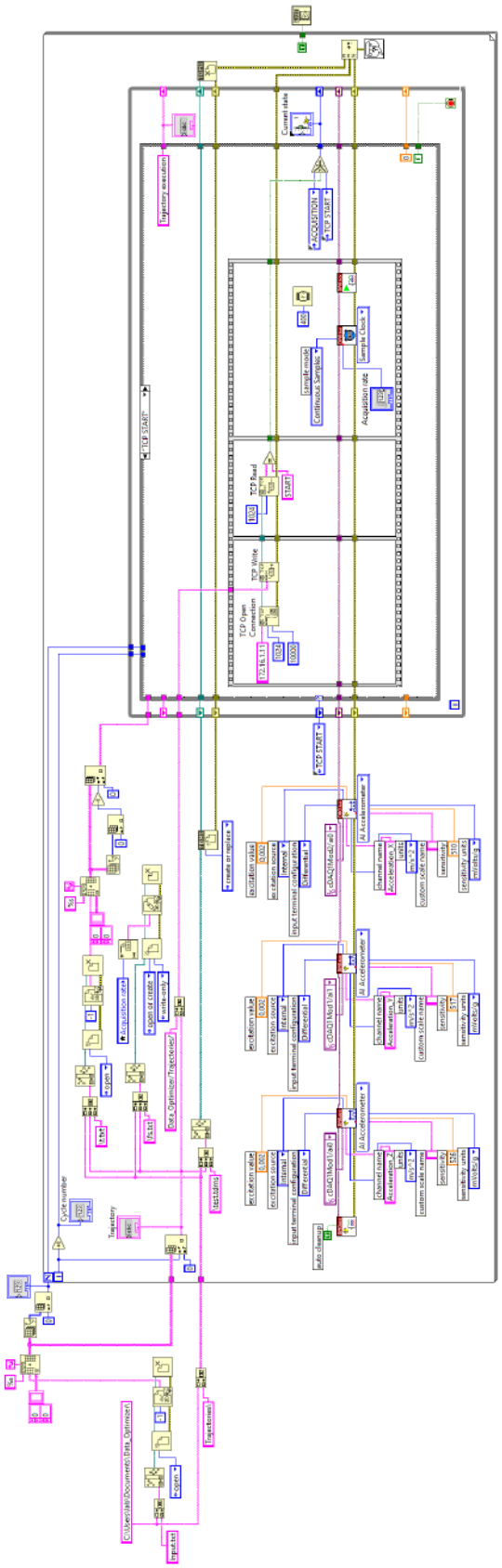


Figure 6.2 Overview of the block diagram of the LabView application used for data acquisition.

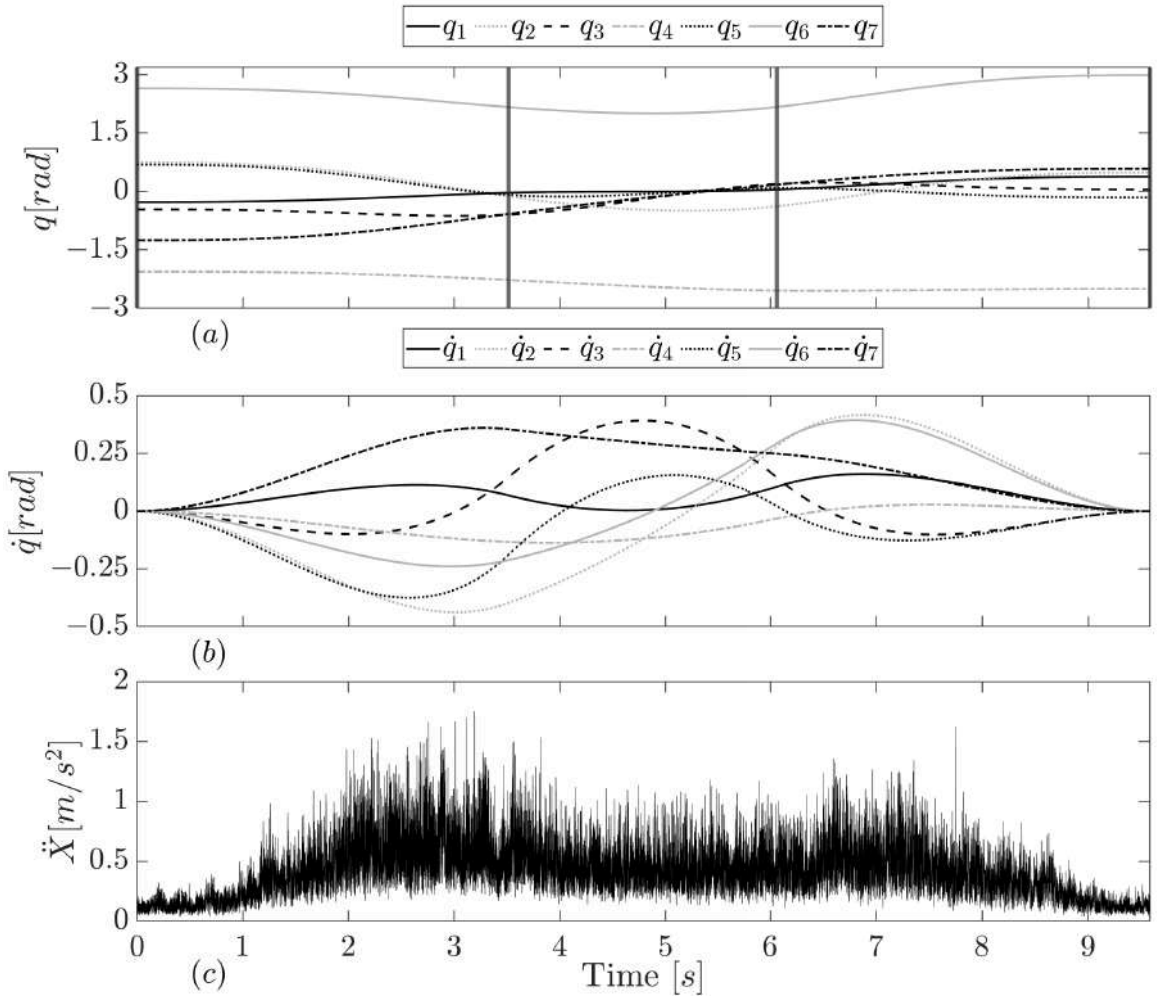


Figure 6.3 Example trajectory: joint position, velocity and absolute acceleration signal for $\alpha = 0.5$. Vertical solid lines indicate target via-points.

Figure 6.4(a) illustrates the trend of T and J as functions of α. As α increases, T decreases with an inflection point around $\alpha = 0.5$, whereas J increases following a parabolic trend. This reflects the shift from jerk-minimizing trajectories to minimum-time trajectories according to Eq. (6.1). Figure 6.4(b) shows the inverse relationship between J and T, highlighting the inherent trade-off.

Finally, Fig. 6.4(c) shows the absolute RMS acceleration $\ddot{X}_{\text{rms}}$ as a function of the total execution time T across three repeated tests. RMS acceleration is not only inversely proportional to T but also closely correlated with the integral of squared jerk J. The three repetitions demonstrate high repeatability, confirming the reliability of both the experimental setup and the acceleration measurements.

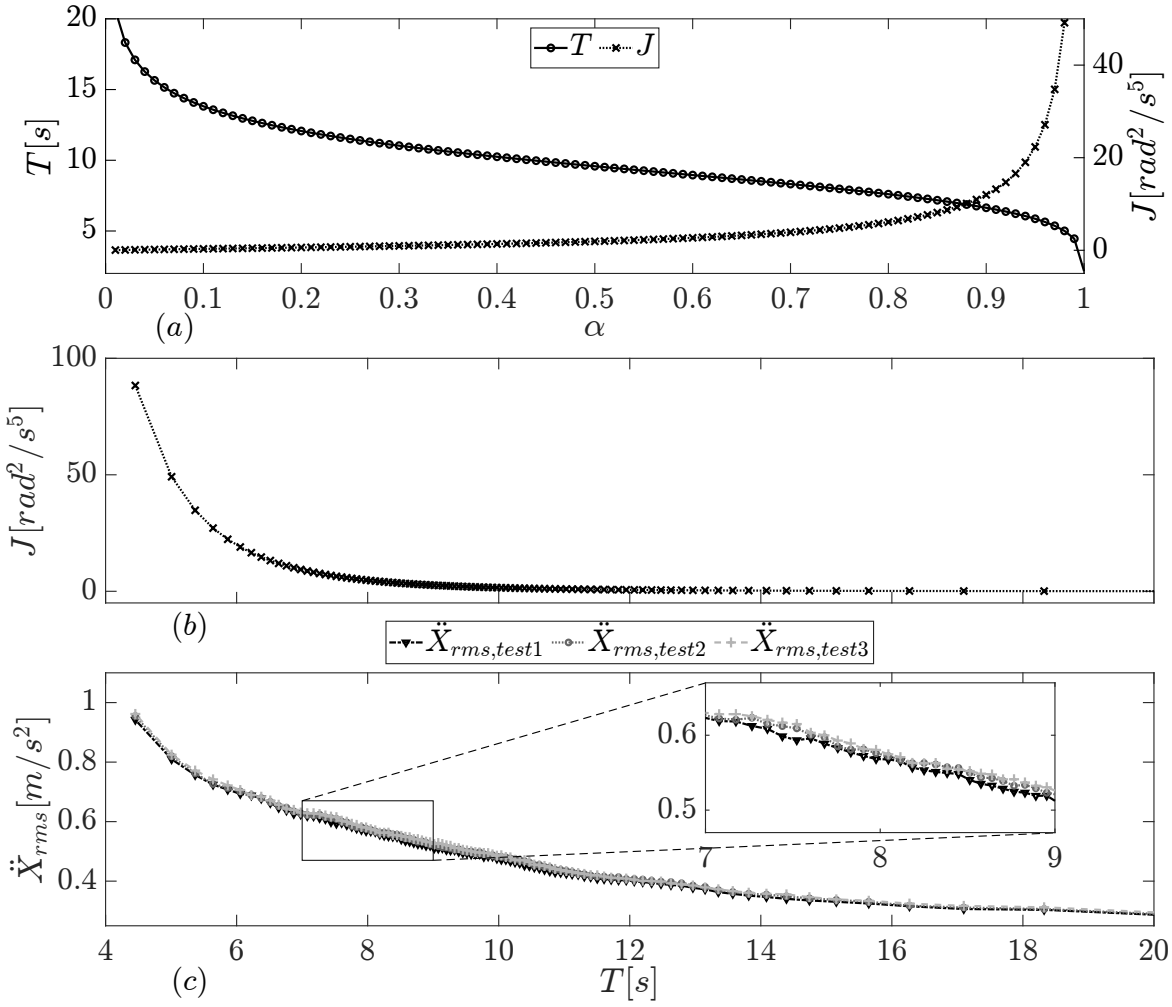


Figure 6.4 (a) T and J vs. α; (b) J vs. T; (c) $\ddot{X}_{\text{rms}}$ vs. T.

6.5 Summary

An experimental setup for testing time-jerk optimal trajectories for robotic manipulators has been presented. The setup includes a Franka Emika robot with 7-DOFs equipped with three accelerometers to measure end-effector oscillations. Smooth trajectories were generated using a "434" spline planning algorithm, optimized via a mixed cost function combining total execution time and joint jerk minimization.

Experimental results demonstrate good agreement with numerical predictions, confirming the feasibility of the proposed method in generating smooth and efficient trajectories while ensuring high accuracy and repeatability of measurements. Future work will focus on exploiting robot redundancy in time-jerk trajectory planning and enabling real-time computation of minimum-jerk trajectories for collaborative robotics, following the approach in Scalera et al. (2022).

Chapter 7

Planning optimal minimum-jerk trajectories for redundant robots

7.1 Proposed Approach

The approach proposed in this chapter for minimum-jerk trajectory planning of redundant robots is based on parameterizing the trajectory using the time intervals between consecutive waypoints to be visited by the manipulator in a fixed sequence. With recpect of the approach proposed in Chapter , this method focuses on the kinematic redundancy of the robot to emphasizes the jerk reduction, optimizing the joint configuration along the trajectory waypoints.

The primary objective is to reduce mechanical vibrations induced by robot motion. This is achieved by minimizing the jerk of the end-effector, a well-established strategy in the literature Barre et al. (2005); Gasparetto et al. (2011); Gasparetto and Zanotto (2007, 2008).

The motion planning procedure optimizes both the vector of time intervals h between consecutive waypoints and the vector of positions q^* of a selected redundant joint at each waypoint. The approach is structured in three optimization stages:

- **Stage 1:** Optimize the vector of time intervals h to reduce both the total trajectory time and the end-effector jerk, as in Gasparetto and Zanotto (2007);
- **Stage 2:** Further reduce end-effector jerk by optimizing the position of a selected joint q^* at each waypoint of the robot path;
- **Stage 3:** Refine the trajectory by re-optimizing the vector h to reduce a weighted sum of total time and end-effector jerk, as in Stage 1.

The Stage 1 optimization considers the following cost function, which combines total time and the integral of squared end-effector jerk along the trajectory $(q, \dot{q}, \ddot{q}, \dddot{q})$:

$$\min_{h} \quad \alpha \sum_{i=1}^{m-1} h_i + (1-\alpha) \sum_{n=1}^{6} \int_0^T \dddot{X}_n^2(q, \dot{q}, \ddot{q}, \dddot{q}) \, \mathrm{d}t \tag{7.1}$$

subject to the kinematic and dynamic constraints:

$$\begin{cases} q_{\min} < q < q_{\max} \\ |\dot{q}| < \dot{q}_{\max} \\ |\ddot{q}| < \ddot{q}_{\max} \\ |\dddot{q}| < \dddot{q}_{\max} \\ |\tau| < \tau_{\max} \\ |\dot{\tau}| < \dot{\tau}_{\max} \end{cases} \tag{7.2}$$

Here, $\alpha \in [0,1]$ balances the contributions of total trajectory time and end-effector jerk, allowing the user to select a desired trade-off. If the total trajectory time T is fixed, the cost function reduces to the minimization of the end-effector jerk:

$$\min_{h} \quad J = \sum_{n=1}^{6} \int_0^T \dddot{X}_n^2(q, \dot{q}, \ddot{q}, \dddot{q}) \, \mathrm{d}t \tag{7.3}$$

The six-component vector of end-effector jerk is computed as:

$$\dddot{X} = J \dddot{q} + 2\dot{J}\ddot{q} + \ddot{J}\dot{q} \tag{7.4}$$

where J is the robot Jacobian and $\dot{J}$, $\ddot{J}$ are its first and second derivatives (see Appendix A.1).

Joint torques are computed using the robot dynamics:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + f_c \mathrm{sign}(\dot{q}) + g(q) \tag{7.5}$$

where $M(q)$ is the mass matrix, $C(q,\dot{q})\dot{q}$ accounts for Coriolis/centrifugal terms, F_v and f_c model viscous and Coulomb friction, and $g(q)$ represents gravity. Torque rates $\dot{\tau}$ are computed as incremental ratios. Alternative friction models and uncertain dynamics formulations are available Fabris et al. (2024a); Giusti and Althoff (2017); Scalera et al. (2024b).

Trajectories are parameterized using a 434 spline, i.e., a third-degree polynomial between most waypoints, and a fourth-degree polynomial at the first and last segments Cook and Ho (1984). For a single joint between waypoints P_k and P_{k+1} ($2 \leq k \leq m-2$):

$$F_k(t) = B_{k,1} + B_{k,2}t + B_{k,3}t^2 + B_{k,4}t^3 \tag{7.6}$$

For the first and last segments ($k = 1$ and $k = m-1$):

$$F_k(t) = B_{k,1} + B_{k,2}t + B_{k,3}t^2 + B_{k,4}t^3 + B_{k,5}t^4 \tag{7.7}$$

The coefficients ensure continuity of position, velocity, and acceleration; jerk continuity is not guaranteed.

In Stage 2, the joint position vector q^* of a selected redundant joint is optimized at each waypoint to further reduce end-effector jerk:

$$\min_{q^*} \quad \sum_{n=1}^{6} \int_0^T \dddot{X}_n^2(q, \dot{q}, \ddot{q}, \dddot{q})\, \mathrm{d}t \tag{7.8}$$

subject to the constraints in (7.2). The robot configuration is determined through inverse kinematics after assigning q^* at each waypoint, allowing optimization of the trajectory smoothness.

7.2 Experimental Setup

The proposed approach is implemented on a Franka Emika Panda arm with 7-DOFs, available at the Mechatronics and Robotics Lab of the University of Udine (Italy), as shown in Fig. 6.1 b. This intrinsically redundant manipulator has a reachability of 855 mm and a maximum payload of 3 kg. The robot is controlled via ROS Melodic Morenia and Python 3.6 on a workstation running Ubuntu 18.04 LTS with an Intel i5-10600k CPU and 32 GB RAM. Trajectories are sent to the robot by specifying sequences of joint positions, velocities, and accelerations through a custom Python script. The Franka Emika arm is suitable not only for trajectory planning research but also for collaborative robotics applications, thanks to its lightweight design, competitive payload-to-weight ratio, and capability to provide joint torque measurements Haddadin et al. (2022).

To measure the end-effector vibrations during the experiments, three PCB piezoelectric mono-axial accelerometers are mounted on a custom aluminum flange (Fig. 6.1c) Lozer et al. (2023). The accelerometers are oriented along three orthogonal planes to capture vibrations

along the x, y, and z axes. Signals from the accelerometers are acquired using two NI9218 C Series modules connected to a cDAQ 9174 chassis and sent to a second workstation running Windows 10 with identical specifications. Data acquisition is performed in LabVIEW 2021 using a state-machine-based Virtual Instrument, synchronized with the robot trajectory via a trigger signal. Accelerometer data are sampled at 10 kHz. Fig. 6.1 illustrates the experimental setup and data acquisition architecture.

The measured accelerations are used to validate the proposed approach, showing reduced root mean square (RMS) and absolute vibration values compared to the reference case. Direct measurement of jerk is challenging, and numerical differentiation of noisy acceleration signals is critical Barre et al. (2005).

The optimization problems in (7.1) with (7.2) are solved in Matlab using *fmincon* with a sequential quadratic programming (SQP) algorithm, with a maximum of 500 iterations. Stage 2 optimization in (7.3) with (7.2) is solved using a genetic algorithm, with a population size ten times the number of waypoints and 50 generations. The inverse kinematics of the redundant robot is solved using the *ikcon* function from the Matlab Robotics Toolbox, which uses *fmincon* with an active-set algorithm to find the optimal robot configuration satisfying the desired end-effector pose. The dynamic model from Gaz et al. (2019) is used to verify joint torques as in (6.3), and kinematic/dynamic limits are taken from GmbH (2017).

The trajectory used to validate the approach is a pick-and-place motion defined by four waypoints in the Cartesian space (Fig. 7.1), with initial and final joint velocities set to zero. Desired end-effector poses $[x, y, z, \varphi, \vartheta, \psi]$ at each waypoint are reported in Tab. 7.1. Roll-Pitch-Yaw angles are used for orientation, with the reference frame origin at the robot base center. The orientation is kept constant across waypoints to simplify analysis of linear accelerations, but the approach can handle variable orientations. Orientation constraints are imposed only at waypoints; the robot can vary orientation between points within its workspace. Inverse kinematics is applied independently of intermediate orientations.

Table 7.1 Desired poses of the robot at each waypoint of the pick-and-place trajectory.

waypoint	x [m]	y [m]	z [m]	φ [rad]	ϑ [rad]	ψ [rad]
1	0.40	-0.40	0.10	π	0	0
2	0.40	-0.30	0.40	π	0	0
3	0.40	0.10	0.40	π	0	0
4	0.40	0.20	0.10	π	0	0

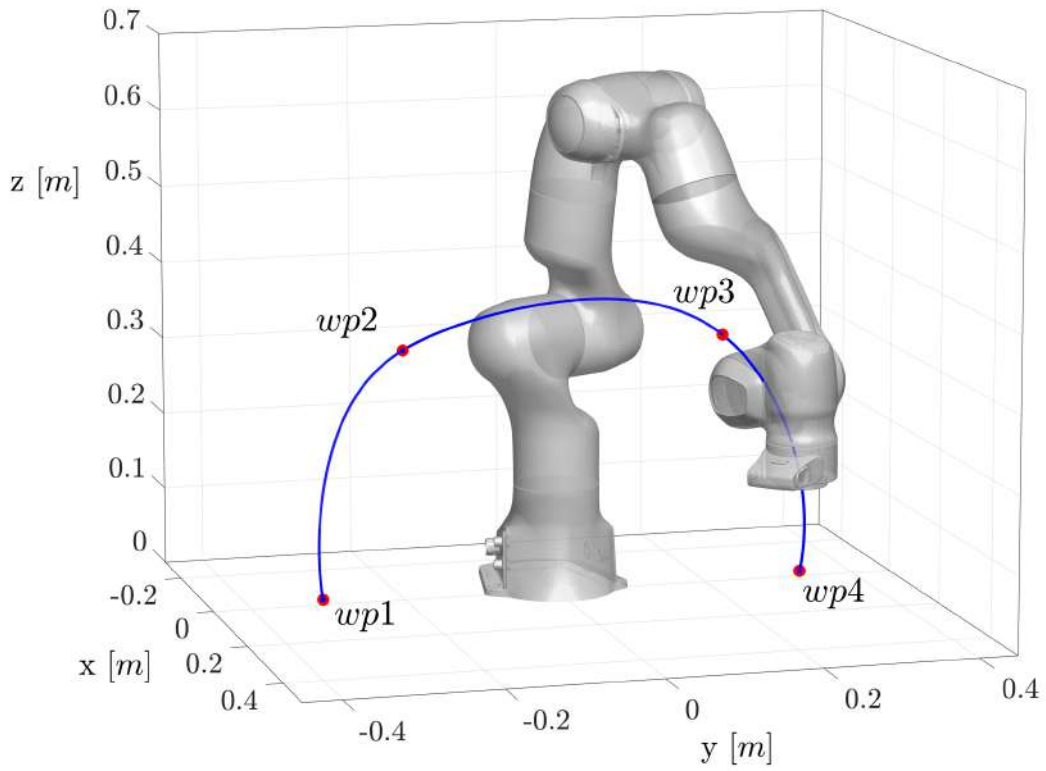


Figure 7.1 waypoints of the pick-and-place task (red markers) and example of the 3D path of the robot (blue curve).

7.3 Experimental Results

The proposed approach is first evaluated numerically using 100 tests where the parameter α is varied from 0.01 to 1 in constant steps, exploring a wide range of possible trajectories. All three optimization stages are applied for each test. These tests are repeated seven times, once for each joint to evaluate the effect of the selected joint position q^* in Stage 2.

A separate set of tests is performed with a fixed total pick-and-place time of $T = 5$ s, applying Stage 1 and Stage 3 optimizations as in (7.3). Stage 2 is again repeated seven times to study the impact of the chosen joint on performance.

Experimental validation is performed with 100 trials on the Franka Emika Panda equipped with accelerometers, following the procedure in Sect. 7.2. For fixed-time experiments, only trajectories corresponding to the best-performing joints in numerical simulations are executed to limit data acquisition.

In both numerical and experimental tests, Stage 1 optimization (minimum time-jerk) serves as a reference case. The complete three-stage optimization, including both optimal time intervals h and joint positions q^*, is compared against this reference. This reference

corresponds to the approach proposed by Gasparetto and Zanotto Gasparetto and Zanotto (2007).

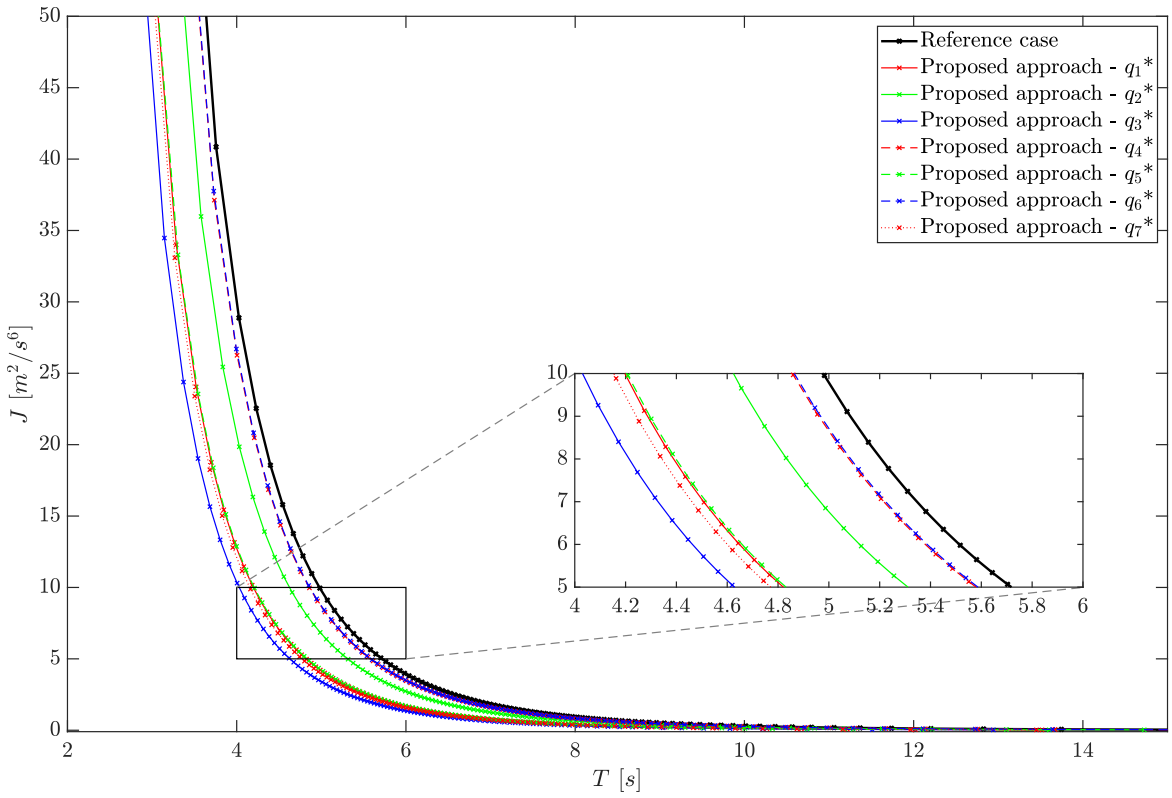


Figure 7.2 Numerical results: time-jerk Pareto front for the optimal trajectory planning.

Figure 7.2 shows the Pareto front for the time-jerk optimal trajectory planning obtained from the 100 numerical simulations by varying the parameter α. Each point in the plot represents a single test, defined by a pair of total time and jerk values (T, J). The black curve corresponds to the reference case (Stage 1, as in Gasparetto and Zanotto (2007)), while the colored curves show the results obtained with the proposed three-stage optimization, by selecting and optimizing the positions of different robot joints in Stage 2. As observed in Fig. 7.2, all colored curves lie below the reference case, indicating that the three-stage optimization consistently achieves better performance than the single time-jerk optimization. In particular, optimizing joints 4 or 6 (red and blue dashed curves) results in limited jerk reduction, whereas optimizing joint 3 (blue continuous curve) yields the most significant improvement in the time-jerk trade-off.

Table 7.2 reports numerical results for the root-mean-square (RMS) and maximum values of the robot end-effector accelerations, along with the integral of the squared jerk J, for selected values of α when optimizing joint 3. The table clearly shows that both

acceleration and jerk increase with increasing α, reflecting the transition from minimum-jerk to minimum-time optimization.

Table 7.2 Numerical results: RMS and maximum values of the accelerations of the robot end-effector, and integral of the squared jerk J for different values of α when optimizing joint 3.

α	$\ddot{x}_{\text{rms}}$ [m/s^2]	$\ddot{y}_{\text{rms}}$ [m/s^2]	$\ddot{z}_{\text{rms}}$ [m/s^2]	$\ddot{X}_{\text{rms}}$ [m/s^2]	$\ddot{x}_{\text{max}}$ [m/s^2]	$\ddot{y}_{\text{max}}$ [m/s^2]	$\ddot{z}_{\text{max}}$ [m/s^2]	$\ddot{X}_{\text{max}}$ [m/s^2]	J [m^2/s^6]
0.01	0.019	0.018	0.031	0.041	0.065	0.043	0.056	0.081	0.029
0.2	0.096	0.088	0.154	0.202	0.325	0.214	0.280	0.401	0.424
0.4	0.156	0.144	0.252	0.330	0.531	0.350	0.458	0.657	0.961
0.6	0.235	0.216	0.378	0.495	0.796	0.525	0.686	0.984	1.888
0.8	0.383	0.352	0.618	0.808	1.300	0.857	1.121	1.607	4.279
1	5.252	5.504	7.333	10.566	12.874	12.363	15.882	18.425	317.170

Figure 7.3 presents the numerical results for a fixed pick-and-place time of $T = 5$ s. Fig. 7.3a shows Stage 2 results (optimization of joint positions), highlighting that optimizing one joint significantly reduces the jerk compared to the reference case, except for joints 4 and 6, which yield minimal reduction. For example, optimizing joint 3 decreases the squared sum of jerk by 56.05%, while joint 2 achieves 58.24% reduction.

Stage 3 further refines the optimization by jointly adjusting time intervals and joint positions. Fig. 7.3b shows that jerk reduction reaches 61.5% for joint 3 and 66.62% for joint 2 compared to the reference case.

Table 7.3 summarizes the RMS and maximum accelerations and jerk values for all optimized joints. Joints 2 and 3 achieve the best reductions in both acceleration and jerk.

Table 7.3 Numerical results: RMS and maximum values of the accelerations and integral of squared jerk J for different optimized robot joints.

	$\ddot{x}_{\text{rms}}$ [m/s^2]	$\ddot{y}_{\text{rms}}$ [m/s^2]	$\ddot{z}_{\text{rms}}$ [m/s^2]	$\ddot{X}_{\text{rms}}$ [m/s^2]	$\ddot{x}_{\text{max}}$ [m/s^2]	$\ddot{y}_{\text{max}}$ [m/s^2]	$\ddot{z}_{\text{max}}$ [m/s^2]	$\ddot{X}_{\text{max}}$ [m/s^2]	J [m^2/s^6]
Reference case	0.507	0.524	0.659	0.982	1.573	1.105	1.257	2.086	13.140
Proposed - q_1^*	0.462	0.418	0.617	0.877	1.545	0.890	1.278	1.986	6.42
Proposed - q_2^*	0.331	0.448	0.464	0.725	1.041	1.113	0.770	1.524	4.39
Proposed - q_3^*	0.410	0.409	0.547	0.797	1.320	0.885	1.122	1.712	5.06
Proposed - q_4^*	0.507	0.524	0.659	0.982	1.574	1.104	1.258	2.087	13.14
Proposed - q_5^*	0.432	0.445	0.531	0.817	1.320	0.993	1.077	1.777	5.96
Proposed - q_6^*	0.505	0.523	0.657	0.980	1.571	1.103	1.253	2.084	13.06
Proposed - q_7^*	0.424	0.438	0.518	0.800	1.277	0.978	1.039	1.732	5.97

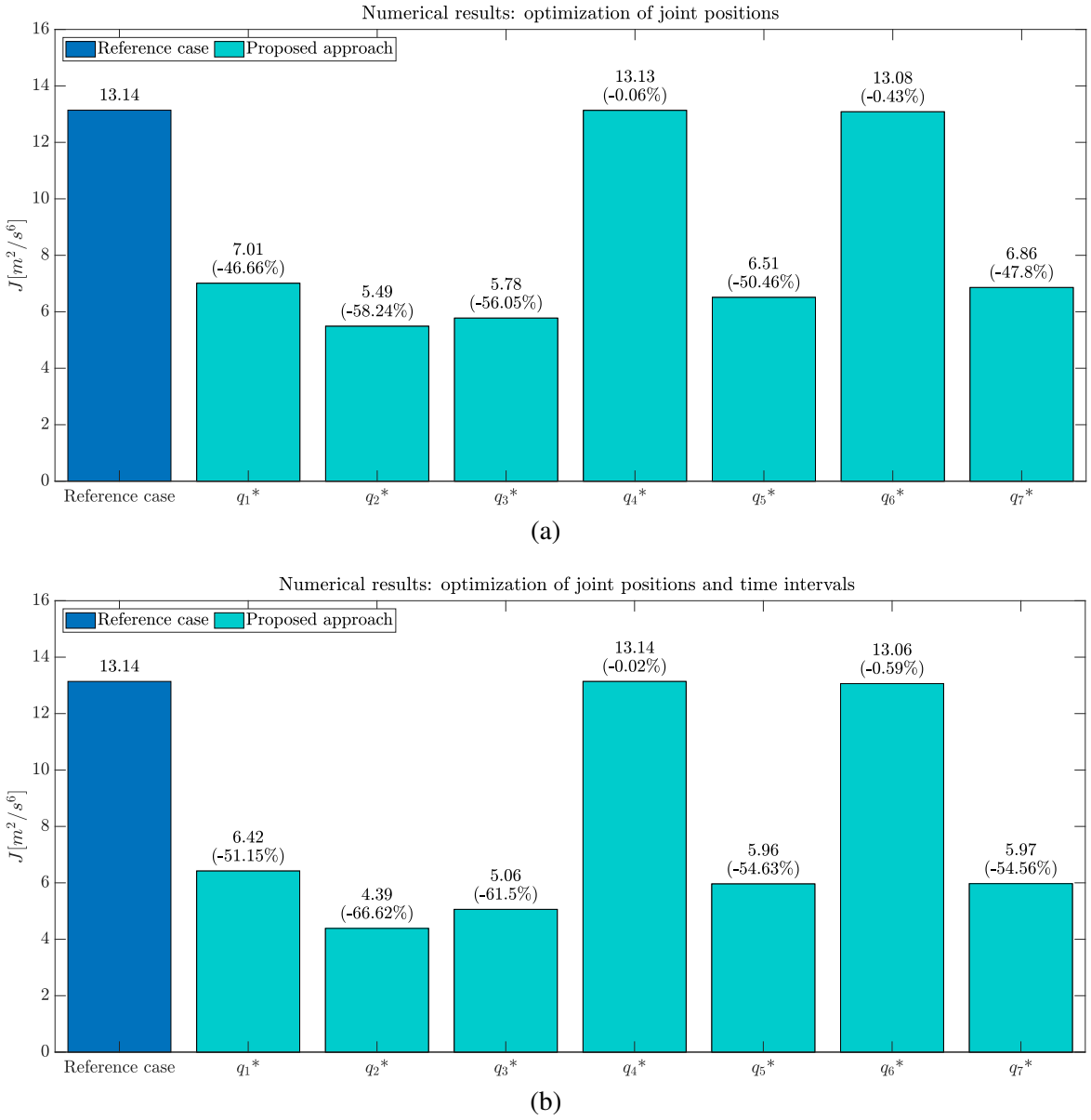


Figure 7.3 Numerical results: (a) RMS of the integral of squared jerk for Stage 2 (joint position optimization) with respect to the reference case; (b) RMS of the integral of squared jerk for Stage 3 (joint positions and time intervals optimization) with respect to the reference case.

Figure 7.4 presents the experimental time-acceleration Pareto front obtained from 100 trials by varying α for the reference case and five optimized joints (1, 2, 3, 5, and 7). Trajectories with optimized joints 4 and 6 were omitted as numerical results indicated minimal improvement. Each point represents the absolute acceleration measured by the PCB accelerometers. The experimental curves consistently lie below the reference case, confirming

that the proposed three-stage optimization effectively reduces measured accelerations in practice.

Table 7.4 reports RMS and maximum accelerations for different α values when optimizing joint 3. The results show a clear increase in acceleration with increasing α, consistent with the numerical findings.

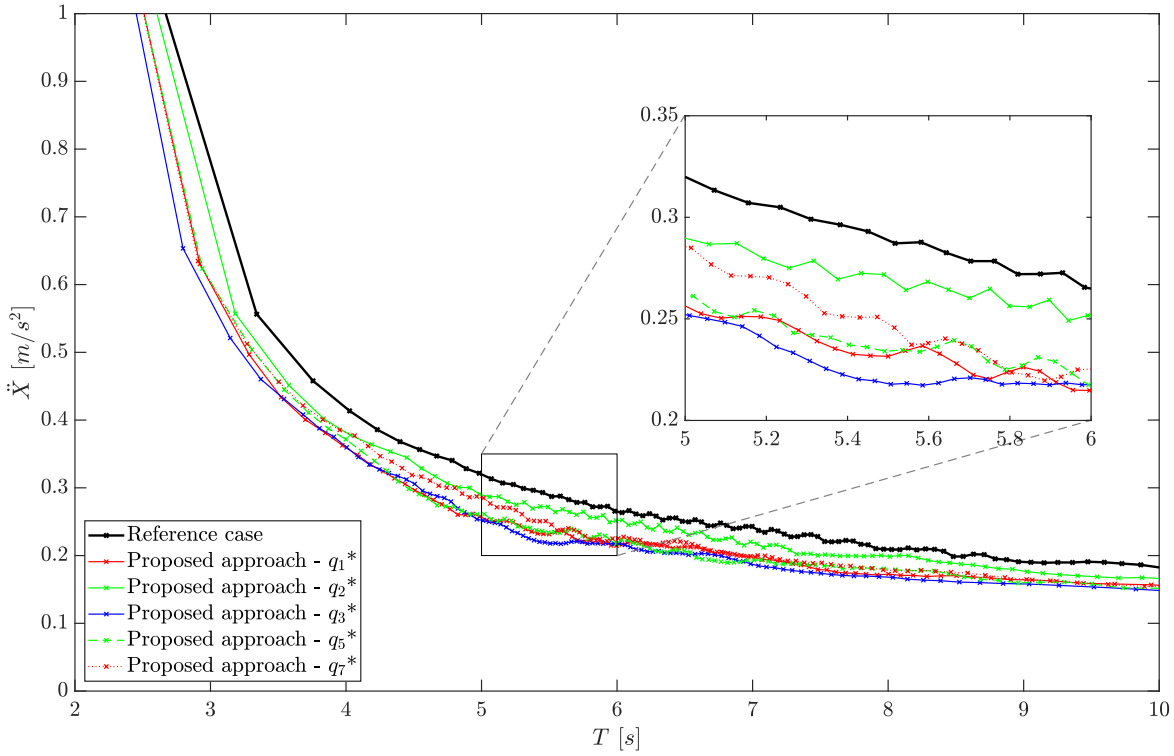


Figure 7.4 Experimental results: time-acceleration Pareto front for the optimal trajectory planning.

Table 7.4 Experimental results: RMS and maximum values of the measured accelerations of the robot end-effector for different values of α with the proposed approach by optimizing the positions of joint 3.

α	$\ddot{x}_{\text{rms}}$ [m/s^2]	$\ddot{y}_{\text{rms}}$ [m/s^2]	$\ddot{z}_{\text{rms}}$ [m/s^2]	$\ddot{X}_{\text{rms}}$ [m/s^2]	$\ddot{x}_{\text{max}}$ [m/s^2]	$\ddot{y}_{\text{max}}$ [m/s^2]	$\ddot{z}_{\text{max}}$ [m/s^2]	$\ddot{X}_{\text{max}}$ [m/s^2]
0.01	0.066	0.085	0.077	0.133	0.304	0.372	0.268	0.435
0.2	0.096	0.110	0.091	0.172	0.466	0.500	0.337	0.533
0.4	0.107	0.139	0.105	0.204	0.464	0.651	0.324	0.667
0.6	0.111	0.145	0.119	0.218	0.470	0.528	0.371	0.598
0.8	0.137	0.193	0.149	0.280	0.476	0.717	0.429	0.794
1	0.459	0.844	0.970	1.365	1.992	2.481	2.529	3.104

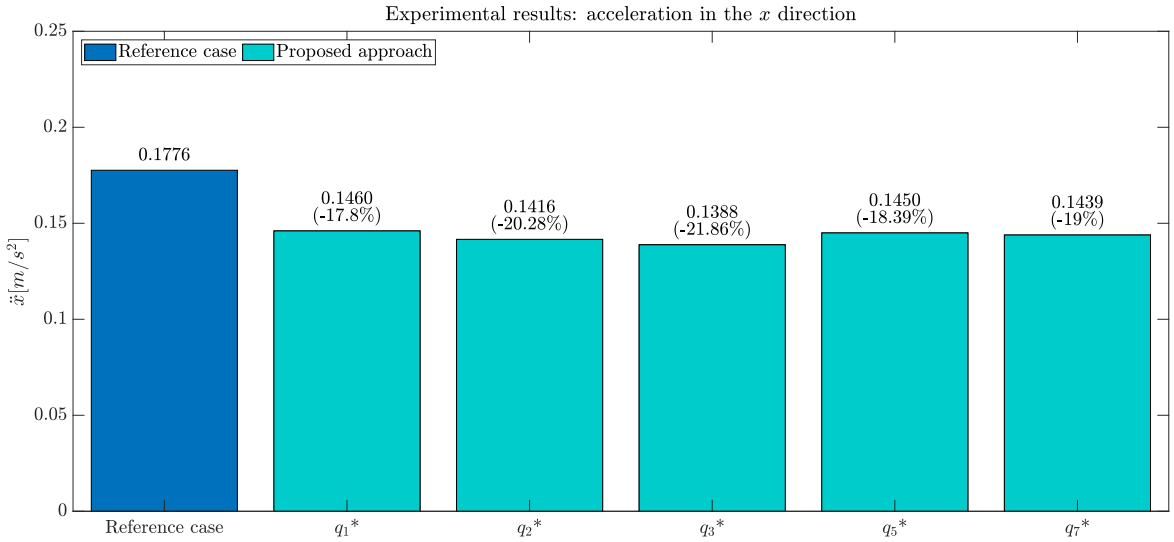


Figure 7.5 Experimental results: measured accelerations in the x direction.

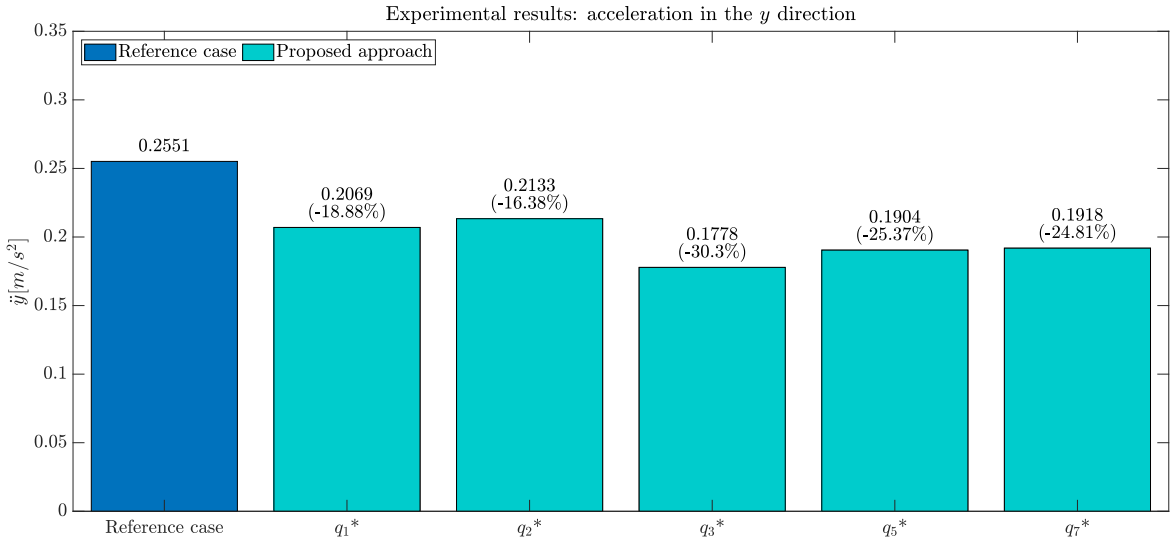


Figure 7.6 Experimental results: measured accelerations in the y direction.

Measured accelerations obtained with the proposed three-stage optimization strategy are presented, where different joint positions are optimized in Stage 2 and the total execution time is set to $T = 5$ s. Specifically, the measured accelerations along the x, y, and z axes of the robot end-effector, as well as the absolute acceleration, are shown in Figures 7.5, 7.6, 7.7, and 7.8, respectively.

The results indicate that optimizing joint 3 (q_3^*) provides the lowest acceleration along the x axis, with a reduction of 21.86% compared to the reference case. Joint 3 also achieves the best performance along the y axis, reducing the measured acceleration by 30.3%. Along the

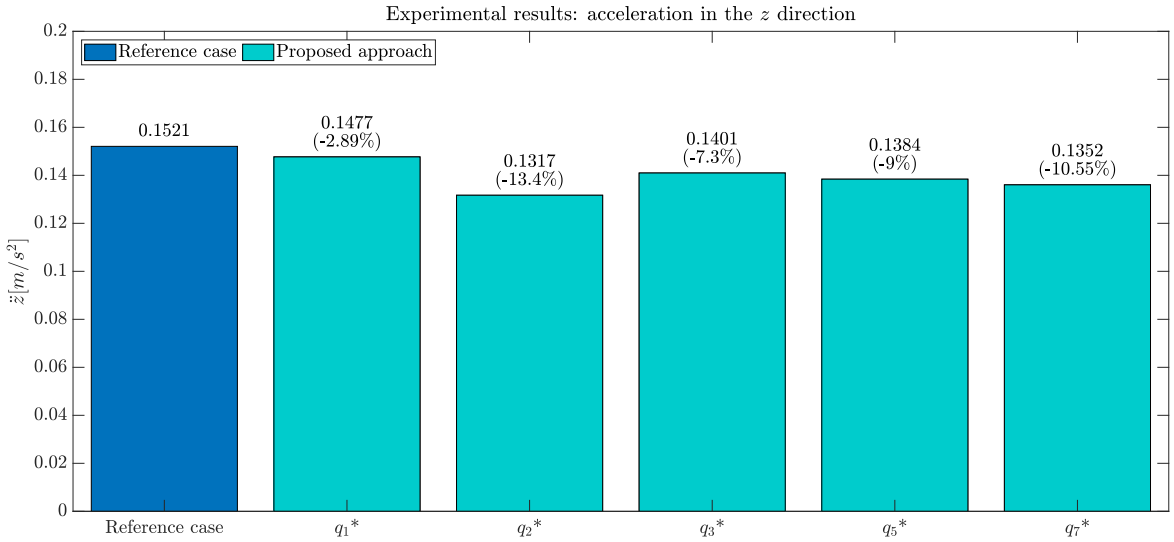


Figure 7.7 Experimental results: measured accelerations in the z direction.

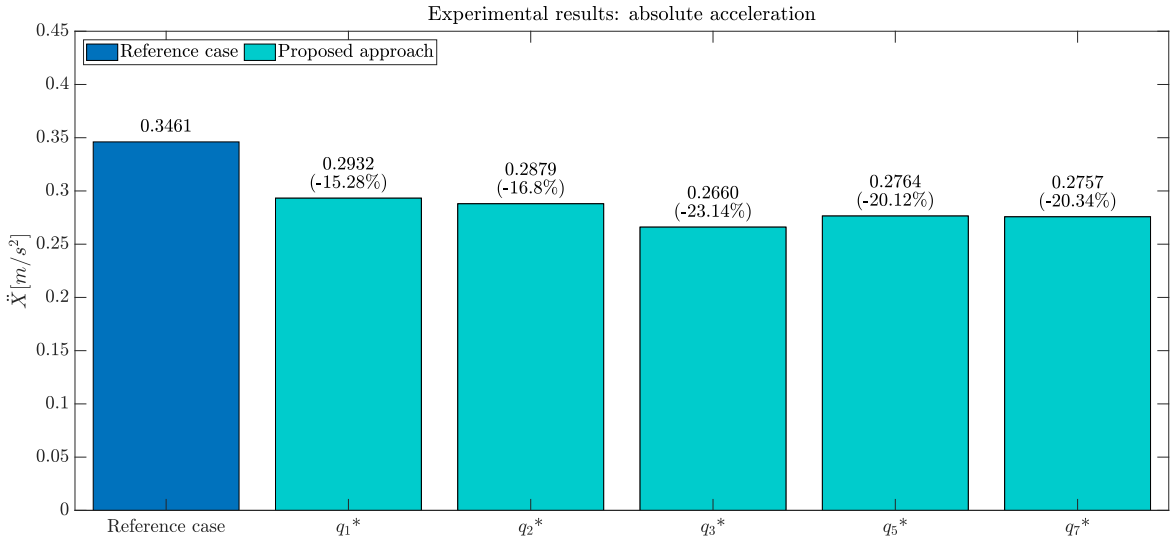


Figure 7.8 Experimental results: measured accelerations in modulus.

z axis, joint 2 yields the lowest acceleration, corresponding to a 13.4% reduction relative to the reference case. Considering the absolute acceleration of the end-effector, optimizing joint 3 leads to the lowest overall values, with a reduction of 23.14% compared to the reference approach.

Table 7.5 summarizes the experimental results in terms of RMS and maximum values of the measured accelerations for the different optimized joints. The data confirm that joints 2 and 3 are the most effective in minimizing end-effector vibrations during the pick-and-place trajectory, demonstrating the practical benefits of the proposed optimization strategy.

Table 7.5 Experimental results: RMS and maximum values of the measured accelerations of the robot end-effector during the tests obtained with the proposed approach by optimizing the positions of the different robot joints.

	$\ddot{x}_{\text{rms}}$ [m/s^2]	$\ddot{y}_{\text{rms}}$ [m/s^2]	$\ddot{z}_{\text{rms}}$ [m/s^2]	$\ddot{X}_{\text{rms}}$ [m/s^2]	$\ddot{x}_{\text{max}}$ [m/s^2]	$\ddot{y}_{\text{max}}$ [m/s^2]	$\ddot{z}_{\text{max}}$ [m/s^2]	$\ddot{X}_{\text{max}}$ [m/s^2]
Reference case	0.178	0.255	0.152	0.346	0.675	1.000	0.468	1.084
Proposed - q_1^*	0.146	0.207	0.148	0.293	0.702	0.788	0.482	0.862
Proposed - q_2^*	0.142	0.213	0.132	0.288	0.500	0.904	0.366	0.962
Proposed - q_3^*	0.139	0.178	0.141	0.266	0.585	0.750	0.451	0.765
Proposed - q_5^*	0.145	0.190	0.138	0.276	0.729	0.815	0.440	0.835
Proposed - q_7^*	0.144	0.192	0.136	0.276	0.671	0.725	0.425	0.796

Table 7.6 Joint positions of the robot for the four waypoints in the reference case.

waypoint	q_1 [rad]	q_2 [rad]	q_3 [rad]	q_4 [rad]	q_5 [rad]	q_6 [rad]	q_7 [rad]
1	−0.161	0.791	−0.571	−2.046	0.830	2.594	−1.336
2	0.229	−0.163	−0.851	−2.275	−0.147	2.163	−0.532
3	0.220	−0.374	0.024	−2.549	0.011	2.175	0.236
4	0.545	0.486	−0.051	−2.505	0.155	2.988	0.347

Table 7.7 Joint positions of the robot for the four waypoints in the proposed approach by optimizing the positions of joint 3.

waypoint	q_1 [rad]	q_2 [rad]	q_3^* [rad]	q_4 [rad]	q_5 [rad]	q_6 [rad]	q_7 [rad]
1	−0.610	0.665	−0.151	−2.099	0.246	2.748	−0.957
2	−0.412	−0.108	−0.227	−2.277	−0.029	2.172	−0.621
3	0.220	−0.374	0.024	−2.549	0.011	2.175	0.236
4	0.545	0.486	−0.051	−2.505	0.155	2.988	0.347

Table 7.8 Time intervals in the reference case and in the proposed approach by optimizing the positions of joint 3.

	h_1 [s]	h_2 [s]	h_3 [s]
Reference case	2.0981	1.1204	1.7815
Proposed approach - q_3^*	1.9283	1.1531	1.9186

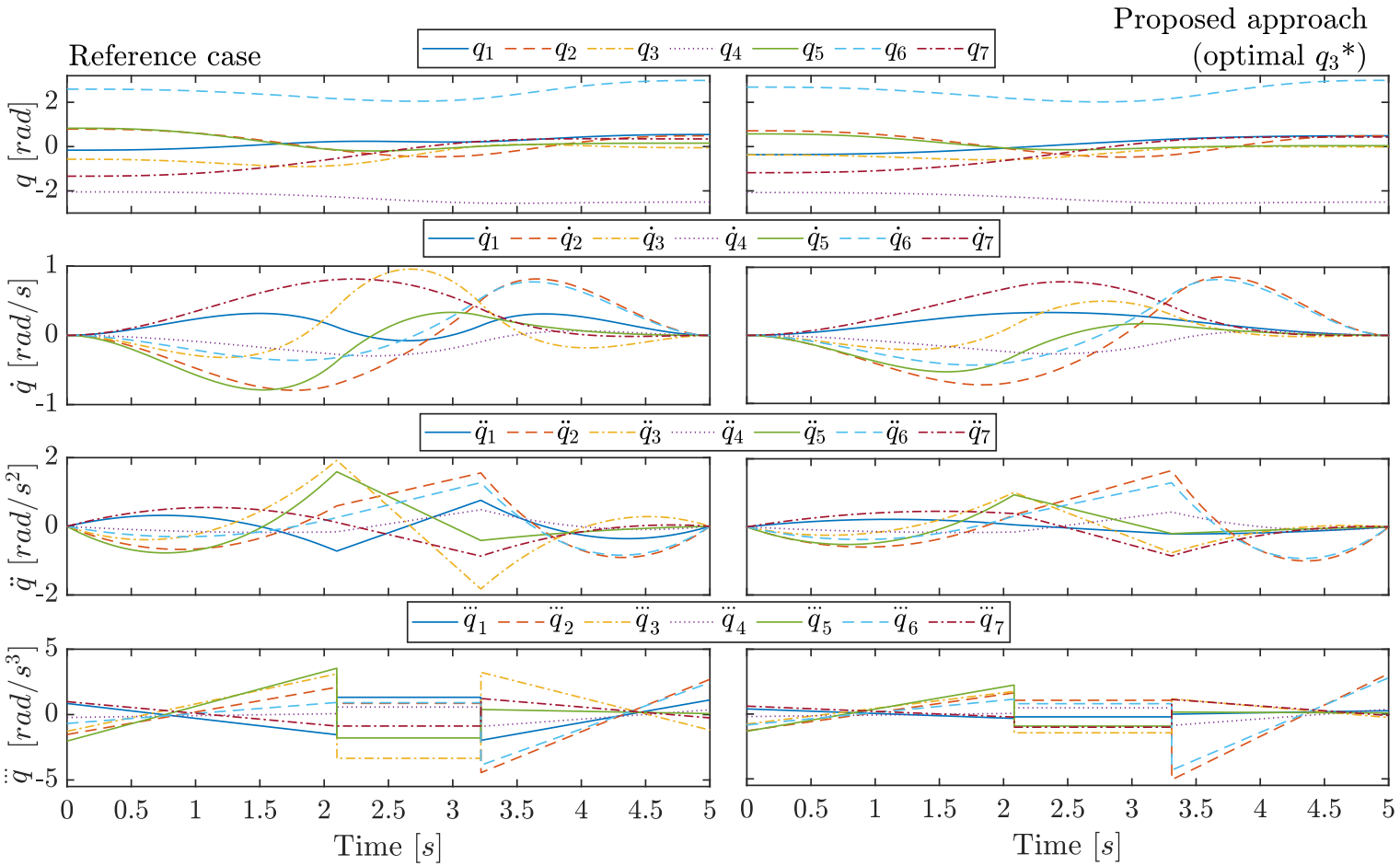


Figure 7.9 Joint positions, velocities, accelerations and jerks for the considered pick-and-place trajectory for the reference case (left), and the proposed approach by optimizing the positions of joint 3 (right).

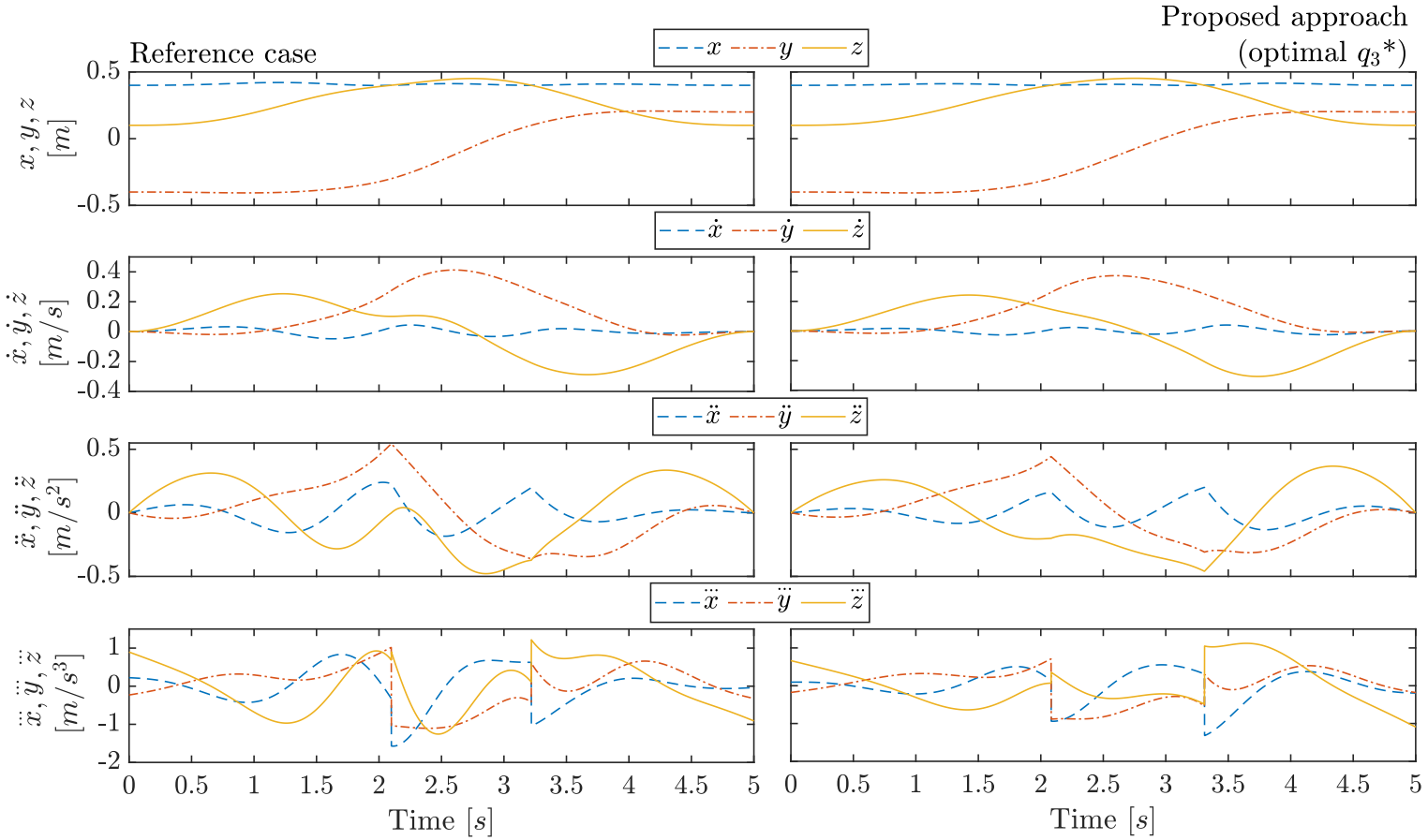


Figure 7.10 End-effector positions, velocities, accelerations and jerks for the considered pick-and-place trajectory for the reference case (left), and the proposed approach by optimizing the positions of joint 3 (right).

For comparison with the reference case (i.e., the minimum-jerk trajectory planning strategy in Gasparetto and Zanotto (2007)), Figure 7.9 shows the joint positions, velocities, accelerations, and jerks along the considered pick-and-place trajectory for both the reference case and the proposed approach with the positions of joint 3 optimized (one of the most promising joints from Fig. 7.8 and Tab. 7.5). Similarly, Figure 7.10 presents the corresponding end-effector positions, velocities, accelerations, and jerks for the same cases. The joint positions of the robot at the four waypoints for the reference case and the proposed approach (with optimized q_3^*) are reported in Tables 7.6 and 7.7, respectively, while the time intervals h for both cases are shown in Table 7.8.

Overall, the proposed approach effectively reduces vibrations in a 7-DOFs redundant manipulator through a novel three-stage optimization procedure that simultaneously considers both time intervals and joint positions for each waypoint along the robot path. By leveraging the redundancy and solving the inverse kinematics of the manipulator at each waypoint, the robot configuration is optimized to minimize the jerk and the induced mechanical vibrations. Optimizing a single joint at each waypoint ensures that the robot configuration is purposefully determined rather than left to chance, offering significant advantages in flexibility and trajectory smoothness compared to conventional 6-DOFs robots.

A key aspect of this methodology is the choice of which joint to optimize during Stage 2. Although selecting the best joint is beyond the scope of this chapter, the numerical and experimental results show that optimizing joints 2 or 3 yields the most significant reductions in end-effector accelerations and jerk. Nonetheless, optimization of the other joints also produces notable improvements relative to the reference case. This highlights that the proposed approach provides a substantial improvement irrespective of the specific joint chosen for Stage 2 optimization.

The method allows jerk minimization without altering the desired pose of the end-effector at each trajectory point, while satisfying all kinematic and dynamic constraints of the robot. By jointly addressing both timing and positioning aspects, the approach enhances trajectory smoothness and precision, reduces vibrations, and potentially extends the lifespan of actuators and mechanical components. These capabilities are particularly important for applications requiring precision, smoothness, and minimal vibrations, emphasizing the value of exploiting redundancy in robots for optimal performance.

7.4 Summary

This chapter presents a multi-phase approach for planning minimum-jerk trajectories in redundant manipulators. The method optimizes both the time intervals between waypoints and the positions of a selected robot joint to further minimize the jerk of the end-effector. The approach was validated through extensive numerical simulations and experimental trials on a 7-DOFs redundant manipulator. The results, supported by accelerometer measurements of induced mechanical vibrations, demonstrate a clear reduction in both acceleration and jerk compared to a state-of-the-art trajectory planning method.

The approach is general and can be applied to other 7-DOFs manipulators, provided their kinematics and dynamics parameters are known. Future work will extend this methodology to more complex robotic systems with different numbers of degrees-of-freedom. For robots with higher redundancy ($N - R > 1$), multiple joint positions could be optimized simultaneously. For non-redundant robots, functional task redundancy could be leveraged. Additionally, future developments may integrate joint positions with redundant spatial task variables, such as end-effector orientation, to achieve optimal minimum-jerk trajectories in diverse robotic applications.

A systematic strategy to pre-select the optimal joint(s) for Stage 2 optimization and to mathematically determine which choices yield the best reduction in jerk and mechanical vibrations will be a focus of future research. This will allow further improvements in trajectory smoothness, ensure a global optimum for the optimization problem, and maximize the performance benefits of the proposed approach.

Part III

Human-like trajectory planning

Chapter 8

Introduction to human-like motion planning

Collaborative robotics is a key topic in Industry 4.0, as the integration of robotic manipulators into manufacturing processes is becoming increasingly important. Unlike traditional industrial robots, modern collaborative robots are designed to work alongside humans without physical barriers. In this context, data acquisition plays a crucial role in analyzing human behavior. For example, in Spitzhirn et al. (2022) a set of platform-based services for ergonomic and productive workplace evaluation is proposed, integrating human motion recordings (via marker-less video capture), digital simulations, and formal task descriptions.

Human-robot collaboration promises greater efficiency and productivity, but it must also ensure a safe working environment Hanna et al. (2022). Safety remains a fundamental requirement whenever humans operate with robots Li et al. (2023a). To guarantee it, measures must be adopted either to minimize injuries in case of collisions Huang et al. (2022); Quiñones et al. (2023), or to avoid unsafe interactions through collision avoidance strategies Merckaert et al. (2022); Wang et al. (2023). Basic guidelines for human-robot interaction are provided by ISO/TS 15066 Standard (2016), which introduces the widely used Speed and Separation Monitoring (SSM) paradigm. Dynamic SSM solutions have been investigated in Yang et al. (2022) using semantic scene information from thermal and depth cameras. Scalera et al. Scalera et al. (2021) extended this concept with dynamic safety zones based on Sphere Swept Lines (SSLs), later experimentally validated in Scalera et al. (2024a). Similarly, Merckaert et al. (2022) proposed a real-time controller for the Franka Emika Panda robot that effectively avoids unpredictable human motions in dynamic environments.

Obstacle avoidance is thus a central aspect of human-robot interaction, directly connected to trajectory planning. Optimizing both paths and trajectories not only prevents collisions

but can also reduce execution time, vibrations, or energy consumption. Several methods have been proposed: in Wang et al. (2019) the TPBSO algorithm (a combination of PSO and BAS heuristics) is used to minimize execution time and joint displacement. Evolutionary algorithms such as GA, SA, and DE have also been applied for path optimization in 6-DOFs manipulators, aiming at torque minimization Baressi Šegota et al. (2020). Graph-based methods are common as well: Malhan et al. Malhan et al. (2022) used Dijkstra's algorithm to minimize path length, while Bottin et al. Bottin and Rosati (2019) generated safe waypoints around a workpiece with the same algorithm. Dai et al. Dai et al. (2020) extended Dijkstra's approach to jerk-optimal trajectories with collision avoidance for 3D printing applications.

More recently, redundancy has been introduced into trajectory optimization. Although redundancy increases dexterity, it also produces infinitely many solutions for the same end-effector pose. Research on redundancy includes energy efficiency enhancement for the UR5 robot Boscariol et al. (2020), null-space constraints for obstacle avoidance Neythalath et al. (2016), and time-jerk trade-offs optimized with genetic algorithms Lozer et al. (2025).

Parallel to these developments, machine learning has emerged as a powerful approach to trajectory optimization. Many methods build on standard algorithms but extend them with data-driven strategies. For example, Neural A* Yonetani et al. (2021) reformulates the A* search into a neural planner that learns to match ground-truth paths. Advanced approaches leverage neural networks and large data models: Huang et al. (2024) integrates RGB-D inputs and language instructions into GPT-4o to generate keypoint constraints for constrained optimization. Reinforcement learning has also been widely adopted, with methods such as TD3 with HER for sample-efficient motion planning Kim et al. (2020), SAC-based approaches for multi-arm systems Prianto et al. (2020), and hybrid RNN-IK models for predicting human arm motion Liu and Liu (2020). Neural methods have also been applied to inverse kinematics, as in Lu et al. (2022a) and Qie et al. (2022), where IK of the Franka Emika Panda was solved using a neural network combined with genetic algorithms.

Inverse kinematics itself remains a key issue in redundant manipulators. Recent works propose analytical solutions for the Panda robot He and Liu (2021); Tittel (2021), highlighting the importance of efficient IK computation in trajectory planning. Building on these ideas, this project takes inspiration from both optimization and machine learning approaches to design a planner that generates human-like trajectories. Unlike other works, the method relies on a custom dataset of recorded human movements to emulate human arm motion. A review of techniques for generating human-like movements in manipulators and humanoids can be found in Gulletta et al. (2020), but this work specifically focuses on replicating human strategies through dataset-driven planning.

To summarize, the main scientific contributions of this section are:

- The development of an approach capable of emulating human arm movements on a redundant robotic manipulator, by exploiting redundancy resolution techniques to generate natural and intuitive trajectories that enhance human-robot collaboration.
- The introduction of a strategy for selecting the robot base orientation, ensuring that the resulting arm motions exhibit human-like characteristics and maintain consistency with natural movement patterns.
- The experimental validation of the proposed framework, carried out through a set of pick-and-place tasks first executed by a human operator and subsequently replicated by a neural network-driven Franka Emika Panda robot, demonstrating the feasibility and effectiveness of the method.

Chapter 9

Human-like moving robot

9.1 Problem Statement

The problem addressed in this chapter is the planning of trajectories similar of those of a human subject in the context of human-robot collaboration. The movements performed by the manipulator may affect his trustability and perception: humans tends to better understand a robot whether its movements are likable Asali and Doshi (2024). A robot behavior which makes it moves in more natural and predictable way, showing human-like movements in performing its tasks, could be considered likable. Human movements, in fact, are inherently smooth and fluent by nature. In contrast, robotic motions generated by standard trajectory planning algorithms tend to be mechanical and, at times, ineffective. Moreover, some tasks are difficult to automate, because of the complex kinematics and the request of a certain level of dexterity, making them difficult to replicate in robotic manipulators using conventional control frameworks.

The project was carried on during my traineeship period abroad at ADMiRE Research center in Villach, Austria.

This chapter aims to solve these issues, proposing an approach to emulate human movements with a collaborative redundant robot. Redundancy is useful in many cases, since it increases dexterity of the robot, but can be also challenging to handle, since can result in an infinite number of kinematic configurations to reach a single pose of the end-effector.

Neglecting the degrees-of-freedom of the fingers, the human arm can be considered as a 7-DOFs system. It is, thus, kinematically redundant. This means that if the hand was fully constrained in terms of position and orientation, forced to assume a specific pose, there would still be one degree of freedom available represented by the elbow position.

Looking at some kind of redundant manipulators, a direct correlation with the human arm structure can be found (Fig. 9.1). This makes the robot kinematics comparable to the human one. Therefore, the manipulator can theoretically emulate the arm configuration and exploit the redundancy to perform the same tasks.

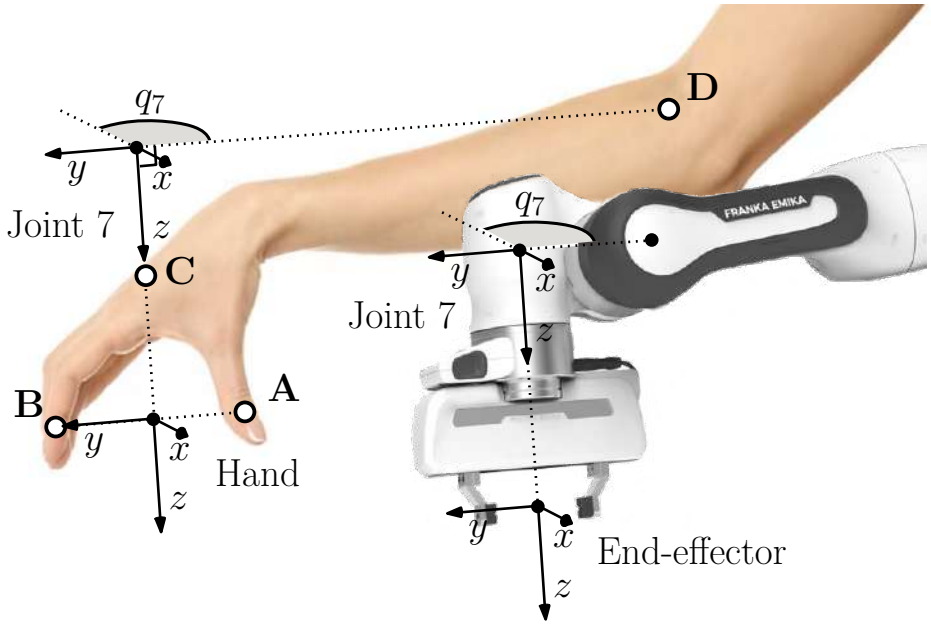


Figure 9.1 Correlation between the reference systems of the robot end-effector and the human hand.

Usually, in trajectory planning, robot configurations can be selected during the inverse kinematics (IK) computation, constraining the end-effector pose along the established path. This task becomes significantly more complex for redundant robots, due to the infinite number of possible solutions. In such cases, analytical solutions are generally unreachable (inside the robot workspace), unless specific constraints are introduced. One common strategy to address this issue is to properly select and optimize the value of a selected redundant joint, effectively reducing the degrees-of-freedom. Depending on the robot configuration and singularities, this can limit the number of solutions to two or four.

Considering a manipulator with 7-DOFs and following the approach described in He and Liu (2021), the seventh joint can be designated as the redundant one. By assigning a value to this joint, redundancy is eliminated, and the solution space becomes finite. This joint can thus be considered as the extra degree of freedom that must be defined to obtain a unique or limited set of IK solutions.

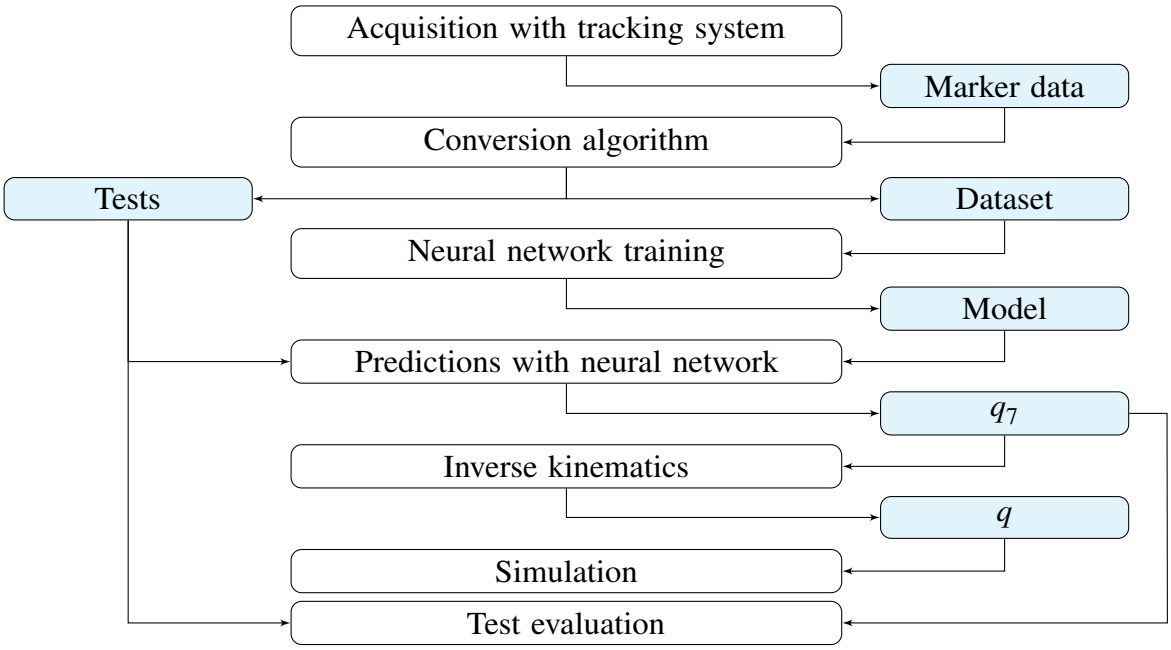


Figure 9.2 Overview of the proposed approach

9.2 Proposed Approach

Determining the optimal robot configuration to replicate human motion and selecting an appropriate redundancy resolution for the IK is the core of this work. The strategy proposed in this chapter combines the analytical method presented in He and Liu (2021) with an artificial neural network to determine the optimal value of a redundant joint of the robot. A neural network trained on a custom-built dataset of real human movements can accurately predict the joint configuration that allows the robot to reproduce human behavior.

9.2.1 Dataset conversion

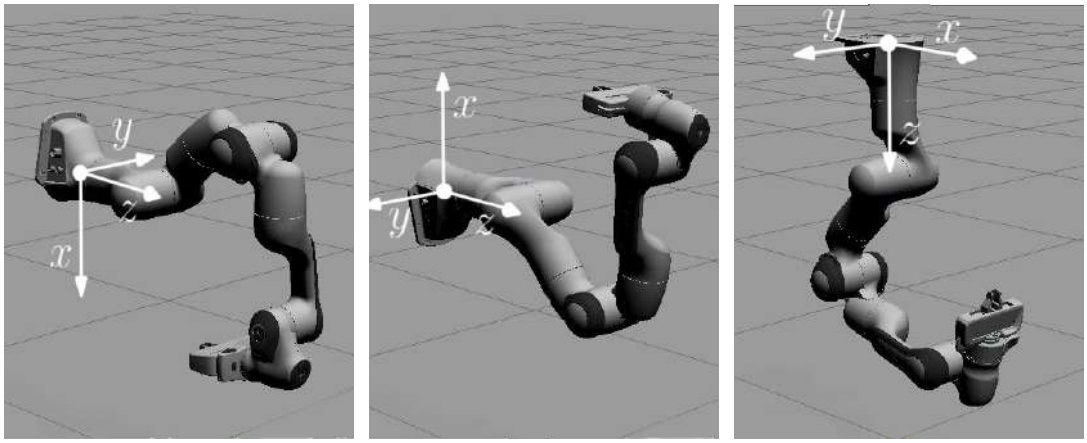
A complete dataset is essential for accurate neural network training. The dataset contains human arm positions recorded during task reproduction with a tracking system. The tracking system acquires x, y, z coordinates of five markers placed on the arm of the human subject. These coordinates define the arm position in the 3D space, but are not directly usable by the neural network. They must be converted into pose matrices, defining position and orientation of the target.

The human arm, modeled as a 7-DOFs system, can replicate the robot kinematics, as it can be seen in Fig. 9.1. Thus, reference frames are placed on the arm to mirror those of the

robot. This conversion links human arm motions to the robotic manipulator ones. There is no need to define the whole kinematic chain, only the base and end-effector frames; others are computed via inverse kinematics.

A custom algorithm converts marker coordinates into usable data: it takes as input the coordinates of the five markers and it returns as output the pose matrix of the end-effector and the value of the 7^{th} joint of the robot. These are the only two data needed by the neural network.

A key step is deciding where to place the robot base for optimal reach. A standard vertical base limits the workspace, so alternative orientations can be explored. Three configurations are tested: two wall-mounted (with different rotations) and one ceiling-mounted. Due to hardware limitations, these custom base orientations are tested only in simulation with a Franka Emika Panda robot.



(a) (b) (c)

Figure 9.3 Examples of orientations of the robot base: horizontal (a), horizontal with x-axis pointing upwards (b) and reversed vertical (c).

The first wall-mounted setup results in limited motion and poor reach below the base (Fig. 9.3a). The second setup improves range slightly, but the robot still could not access lower areas (Fig. 9.3b). The third ceiling-mounted setup offers the best range and human-like workspace (Fig. 9.3c). It allowed the robot to reach most areas used by a human arm on a desk. Thus, the ceiling-mounted orientation is selected in this work. The center of the 2^{nd} joint reference system is assumed to be the shoulder marker. Then, the end-effector pose coordinates is transformed according to the chosen base reference system.

9.2.2 Data computation

Starting from the coordinates of the five markers, it is possible to obtain the pose matrix of the end-effector (EE) and the value of the 7$^{\text{th}}$ joint. To define the orientation of the end-effector and to obtain its pose in the 3D space, it is needed to calculate the quaternion of the reference system fixed on the human. With reference to Fig. 9.1, the center of the end-effector is first calculated as the middle point of the segment linking the two markers placed on the thumb (A) and the middle finger (B), as:

$$O_{\text{EE}} = \frac{A+B}{2} \tag{9.1}$$

Then, the z-axis is placed in the segment between the hand marker (C) and the center, in the opposite direction with respect to the hand:

$$z_{\text{EE}} = \frac{O_{\text{EE}} - C}{||O_{\text{EE}} - C||} \tag{9.2}$$

Instead, the y-axis is placed in the segment between the projection of the finger marker along the z-axis and the finger marker itself:

$$y_{\text{EE}} = \frac{B - (O_{\text{EE}} + z_{\text{EE}}(z_{\text{EE}} \cdot (B - O_{\text{EE}})))}{||B - (O_{\text{EE}} + z_{\text{EE}}(z_{\text{EE}} \cdot (B - O_{\text{EE}})))||} \tag{9.3}$$

Finally, the x-axis is fixed to complete a right-handed triad:

$$x_{\text{EE}} = y_{\text{EE}} \times z_{\text{EE}} \tag{9.4}$$

Once all axes are defined and the position of the center known, it is then possible to build the homogeneous matrix P. The quaternion Q that defines the end-effector orientation in the 3D space is subsequently derived from this matrix.

The second step of the algorithm is the calculation of the position of the 7$^{\text{th}}$ joint. Its value can be interpreted as the degree of rotation of the human wrist with respect to the forearm. It is calculated as the angle formed by the segment between the projection of the elbow marker (D) along the z-axis, as shown in Eq. (9.5), and the elbow marker itself, relative to the previously calculated x-axis with inverse direction, as in Eq. (9.6):

$$O_7 = C + z_{\text{EE}}(z_{\text{EE}} \cdot (D - C)) \tag{9.5}$$

$$q_7 = \cos^{-1}\left(\frac{D - O_7}{||D - O_7||} \cdot (-x_{\text{EE}})\right) \tag{9.6}$$

The value of the 7th joint is corrected to match the standard orientation of the robot's end-effector reference system. Since the simulations uses a Franka Emika Panda robot as the model, according to its official Denavit-Hartenberg parameters, the reference system at the end-effector is rotated by 45° relative to the direction along which the gripper fingers move. Additionally, the positive rotation direction of the human hand corresponds to the negative direction in the robot end-effector. Therefore, it is necessary to correct the value of the 7th joint by inverting its sign and adding a rotation offset, as in Eq. (9.7):

$$q_{7,\text{corr}} = \frac{\pi}{4} - q_7 \tag{9.7}$$

9.2.3 Neural network training

Having obtained all the data needed with the conversion algorithm, it is possible to train the neural network used in this project. The dataset acquired and converted is split into 80% for training and 20% for evaluation. Training uses backpropagation over a defined number of epochs to minimize error. Evaluation tests the model on new data to assess its accuracy. The accuracy is calculated as the percentage of predictions within the 1% of the range of motion of the 7th joint ([-2.8973 rad, 2.8973 rad]) compared to the reference value. So, the training loop described ends outputting a model and its accuracy.

Multiple training loops are run, each with different hyperparameter combinations. At the end, only the model with higher accuracy value is retained. Thus, the best-performing model and its hyperparameters are saved, to be used in the main application to make predictions. The resulting hyperparameters are listed in Tab. 9.1. Finally, a pseudocode of the human-like motion planner proposed is reported in Algorithm 1.

Table 9.1 Hyperparameters of the neural network.

n. epochs	batch size	n. layers	training data	evaluation data
100	1000	9	80%	20%

activation f.	loss f.	optimization f.	learning rate
Tanh	MSELoss	Adam	0.01

1st layer	2nd layer	3rd layer	4th layer	5th layer	6th layer	7th layer	8th layer	9th layer
30	70	60	70	60	55	35	15	5

Algorithm 1 Human-like motion planner

```
markers ← tracking system
for marker in markers do
    if marker from tests then
        P_t, q_7,t ← TEST(marker)
    else if marker from dataset then
        P_d, q_7,d ← DATASET(marker)
    end if
end for
model ← TRAINING(P_d, q_7,d)
for P_t in P_t do
    q_7,NN ← PREDICTION(model, P_t)
    q ← IK(P_t, q_7,NN)
    accuracy ← EVALUATION(q_7,NN, q_7,t)
end for
SIMULATION(q) =0
```

9.3 Experimental Setup

Data are collected using a tracking system, shown in Fig. 9.4, composed of five Optitrack cameras, calibrated and synchronized using the Motive software. Five sticky markers are placed on the tester arm: on the thumb (A), middle finger (B), back of the hand (C), elbow (D) and shoulder (E), as in Fig. 9.1. The tracking system is used for both dataset creation and motion recording. During dataset acquisition, the human tester is asked to make natural, varied arm movements, avoiding unfeasible and difficult poses. In tests recordings, the tester performs some tasks sitting on a desk. These tasks include picking, placing, rotating, flipping, and stacking some small bricks, as it can be seen in Fig. 9.5.

Dataset acquisitions are used for the training of the neural network. Tests recordings are kept separate and used only for final evaluation, to check the accuracy of the predictions made in real case scenarios.

The core of this work is contained in the main program, which, starting from a trajectory in the Cartesian space, gives prediction about the redundant joint and computes the joint trajectory by calculating the inverse kinematics. A workstation running Ubuntu 18.04 is used. The software is written in Python, as the neural network classes, for both training and predictions, and the motion planning server, while the server for IK computation is written in C++. The software for data elaboration and conversion, used for the creation of both tests and dataset, are written in C++ with modules in Python. The fully connected architecture of the neural network is developed using the PyTorch library, version 2.4.1+cpu. The robot control

Figure 9.4 Overview of the setup of the tracking system.

is performed by the mean of the ROS environment, using the Melodic Morenia distribution. After the trajectory is computed in the main program, it is passed to the controller server through TCP protocol and sent to the robot controller via ROS topic. The manipulator is

Figure 9.5 Camera snapshots of the recording of different pick-and-place tests.

simulated in Gazebo, using the default Urdf model provided and the standard joint effort controller.

To evaluate the performance of the proposed approach, the model is tested experimentally. Task executed by the tester are replicated in simulation, and the predictions given by the neural network are compared with the real data. The RMS error between the reference values of the 7^{th} joint, given by the tests acquisitions, and the experimental ones, provided by the application, along the whole trajectory, is calculated as in Eq. (9.8).

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{key}}} \sum_{i=0}^{n_{\text{key}}} (q_{7,i} - q_{7,i}^{\text{ref}})^2} \tag{9.8}$$

where n_{key} is the number of keypoints in the trajectory, $q_{7,i}$ and $q_{7,i}^{\text{ref}}$ are the experimental and reference of the values of the 7^{th} joint on the i^{th} keypoint, respectively.

Then, the grade of accuracy of the model is computed as the RMSE divided by the range of motion of the 7^{th} joint ([-2.8973 rad, 2.8973 rad]), in percentage, as in Eq. (9.9):

$$E = \text{RMSE}/(2 \cdot 2.8973) \tag{9.9}$$

The tests evaluation consists of 7 phases:

- The test is recorded with the tracking system.
- Test data are converted. This operations generates three files: a CSV file with the reference data and two JSON files, one with the end-effector poses and one with the gripper actions.
- The neural network is trained with the dataset, generating the model to be used in the application.
- The application is executed with test data. For every waypoint, the neural network predicts the value of the 7^{th} joint with respect of end-effector poses in the JSON file.
- The inverse kinematics algorithm calculates the joint configurations, taking as input the end-effector poses and the values of the 7^{th} joint.
- The trajectory planner sends the message to simulate the motion in Gazebo, using ROS topics for the communication.
- Finally, the neural network predictions are compared with reference data in the CSV file.

9.4 Experimental Results

A total of four simulation tests were conducted using recorded pick-and-place tasks performed by a human tester. These tests involve picking and placing two or four small bricks at varying positions and heights to analyze different arm movements. For each test, performance metrics such as RMSE, error, computation time for both neural network and inverse kinematics, and the number of successful solutions relative to total waypoints are reported in Tab. 9.2. The model used for the experimental evaluation achieved a result of 93% of accuracy during the training.

The error between the reference and the experimental data is quite small, meaning that the proposed approach is able to give accurate predictions. The computational time taken by

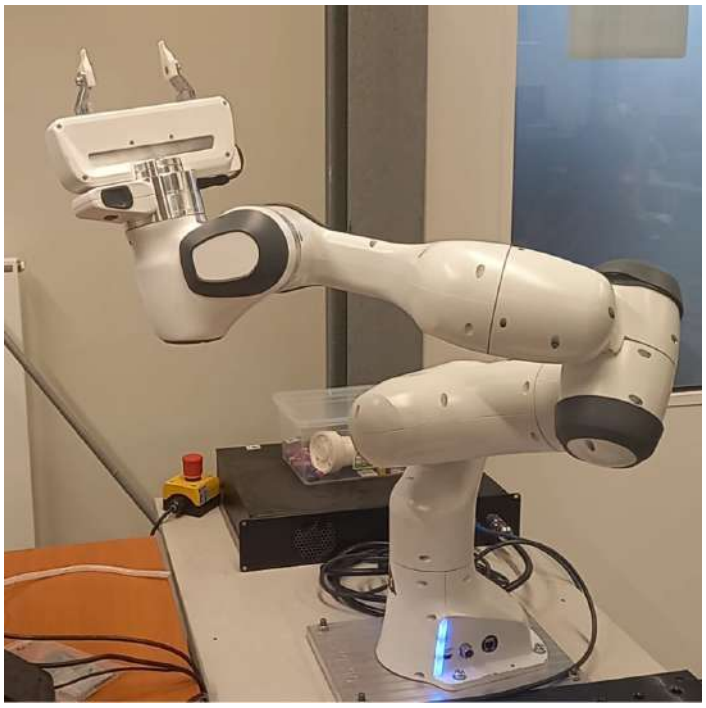


Figure 9.6 The Franka Emika Panda robot used for the experimental validation.

the neural network and the inverse kinematics is overall acceptable, taking into account that the number of keypoint elaborated is above 150.

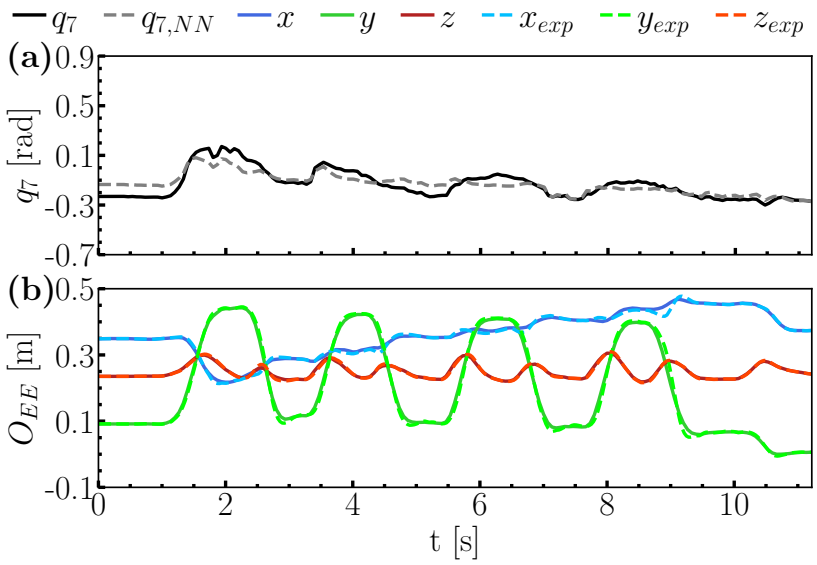


Figure 9.7 Results of the test $n.1$: comparison between reference and experimental values of q_7 (a), comparison between desired and measured O_{EE} trajectory (b).

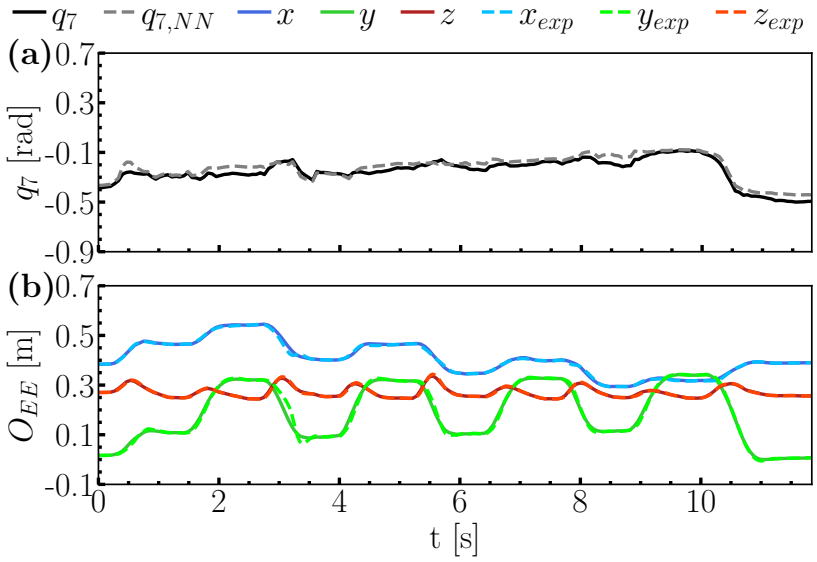


Figure 9.8 Results of the test n.2: comparison between reference and experimental values of q_7 (a), comparison between desired and measured O_{EE} trajectory (b).

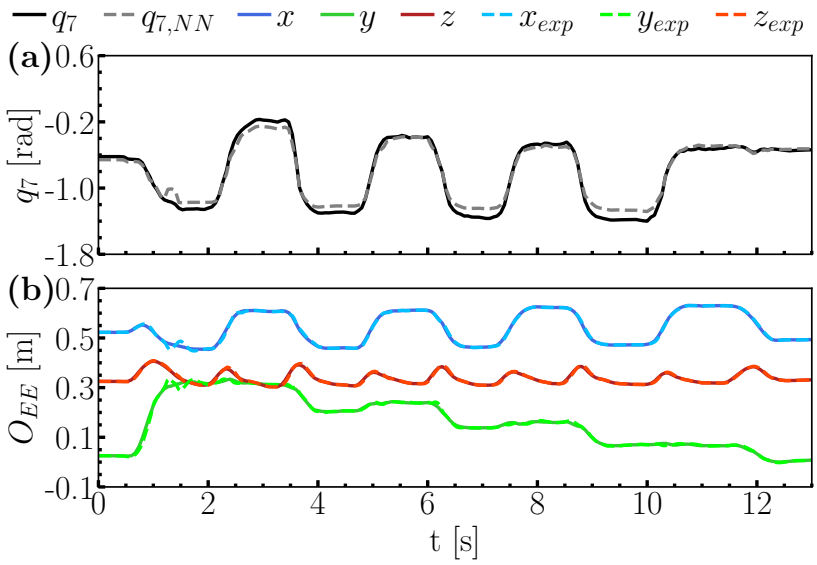


Figure 9.9 Results of the test n.3: comparison between reference and experimental values of q_7 (a), comparison between desired and measured O_{EE} trajectory (b).

In all the tests, the values of the position q_7 of the 7^{th} joint predicted by the neural network overlaps quite well with the ones recorded with the tracking system. The proposed tasks are reproduced with high accuracy by the robot, shown in Fig. 9.6. Due to integrated software limitations, it is not possible to change the orientation of the robot base. Thus, trajectories are executed flipped upside down.

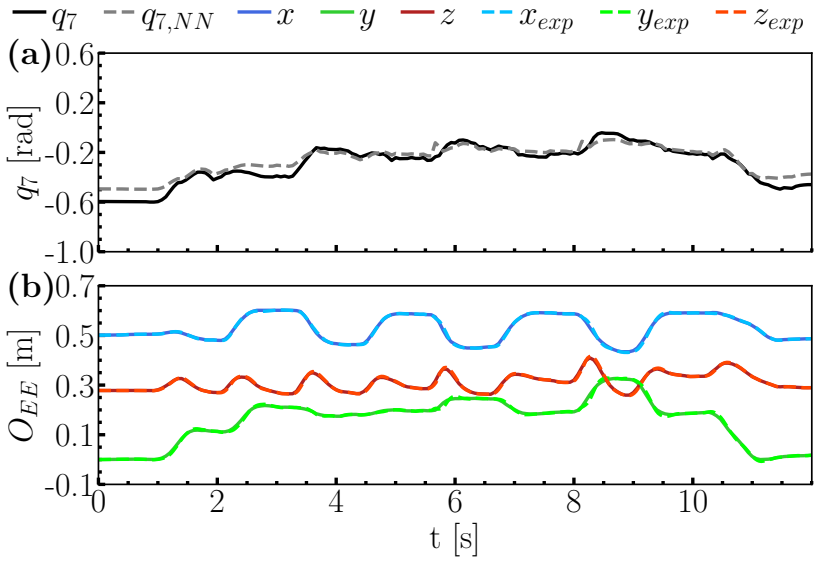


Figure 9.10 Results of the test $n.4$: comparison between reference and experimental values of q_7 (a), comparison between desired and measured O_{EE} trajectory (b).

Table 9.2 Results of the experimental tests.

Test	RMSE [rad]	Error [%]	NN time [s]	IK time [s]	solutions/ waypoints	Fig.
$n.1$	0.058	1.003	0.308	0.044	169/169	9.7
$n.2$	0.039	0.673	0.299	0.041	177/177	9.8
$n.3$	0.066	1.132	0.354	0.055	196/196	9.9
$n.4$	0.053	0.909	0.386	0.040	181/181	9.10

The Cartesian path routed by the robot end-effector coincides with the one created in the tests acquisitions. Thus, taking into account the position of the redundant joint and the end-effector trajectory, the robot faithfully emulates the task performed by the human. Tests $n.1$, $n.2$, $n.3$ and $n.4$ (Fig. 9.7, 9.8, 9.9, 9.10) show a good performance by the neural network in terms of prediction accuracy. Due to the value of the 7^{th} joint obtained by the neural network, the robot occasionally works near the shoulder singularity, switching with high velocity from a configuration to the following one. This sudden movements produce some small vibrations of the end-effector, affecting the smoothness of the trajectory, deviating from the desired path. Anyway, the overall performances of the proposed approach are effective and reliable. Thus, since the predictions are in line with reference data and the generated trajectories are feasible by the robot, we can state that the proposed approach is able to satisfy the requested features and to correctly reproducing human-like movements.

9.5 Summary

In this study, we demonstrated the effectiveness of neural networks in emulating human arm movements on a redundant robotic manipulator. The proposed architecture successfully captured several motion patterns through training on human motion datasets. Results indicate a high degree of accuracy in trajectory reproduction, and the neural network ability to generalize across varying tasks highlights its adaptability.

Compared to traditional control methods, the proposed approach shows improved responsiveness and smoother actuation. Overall, our approach based on a neural network presents a promising approach for achieving human-like dexterity in robotic manipulation. Limitations include sensitivity to robot vibrations, sensor noise, and the need for large training datasets. Future works will focus on increasing dataset size for even better accuracy and improving inverse kinematics and path planning algorithms to resolve possible singularity issues.

Chapter 10

Conclusions

Trajectory planning in collaborative robotics is a multifaceted challenge that demands the integration of different objectives-ranging from human-like motion generation to trajectory optimization and real-time safety assurance. This thesis has explored three critical dimensions of trajectory planning: the emulation of human-like movements, the pursuit of minimum time and jerk optimization, and the implementation of robust collision avoidance strategies.

Collision avoidance remains a foundational requirement for safe operation in shared workspaces. Real-time sensing, environmental mapping, and reactive control algorithms enable robots to detect and respond to obstacles with precision and reliability. The integration of collision avoidance with human-like planning and optimization is a future prospective that might ensure that robots can operate both safely and socially in dynamic environments.

The proposed supervisory controller tracked distances between human and robot bounding volumes and triggered stop trajectories when collisions were possible. Twenty-seven participants followed predefined paths and intentionally entered the robot's workspace. Experiments with Panda robot showed that online scaling of dynamic safety zones significantly enhanced fluency, yielding total task time reductions of 60.7% with the optimal scaling method and 53.0% with the linear search method compared to static zones.

The implementation of the proposed collision avoidance approach in the automation of attaching small components to car rear lamps was studied. Robot pick-and-place trajectories were generated, a custom end-effector was developed, and a safety mechanism was integrated to stop the robot upon potential human contact. The influence of cycle time on collaboration quality and robot energy usage was assessed. Tests on a UR5e robot confirmed that the task can be performed safely and efficiently while satisfying mechanical and timing constraints.

Minimum time-jerk optimization, on the other hand, addresses the need for efficiency and mechanical smoothness. Optimizing trajectories for minimal execution time ensures

high productivity, while minimizing jerk reduces wear on robotic components and enhances comfort in shared environments. These objectives often require solving complex multi-objective optimization problems, balancing the parameters under dynamic constraints.

A multi-phase method for generating minimum-jerk trajectories in redundant manipulators was introduced and tested. The technique jointly optimizes waypoint timing and the configuration of a chosen joint to further reduce end-effector jerk. Numerical simulations and experiments on a 7-DOFs manipulator—supported by accelerometer measurements—show significant reductions in acceleration and jerk compared to a state-of-the-art planner.

The method is generalizable to other 7-DOFs robots, given known kinematic and dynamic parameters. Redundancy was set as an optimization parameter, giving good results in terms of jerk reduction. Future work includes extending the approach to systems with different degrees of redundancy and optimizing multiple joints when higher redundancy is available.

Human-like trajectory planning enhances the intuitiveness and social acceptability of robot behavior, fostering smoother and more natural interactions with human partners. By modeling biomechanical patterns and incorporating principles of predictability, robots can move in ways that align with human expectations, thereby improving trust and collaboration.

The study presented in this thesis shows that neural networks can effectively emulate human arm movements on a redundant robotic manipulator. Trained on human motion data, the proposed architecture accurately reproduces diverse trajectories and generalizes well across different tasks, yielding smoother and more responsive motion than traditional control methods. The approach demonstrates strong potential for achieving human-like dexterity, though it remains sensitive to vibrations, sensor noise, and requires large training datasets. Future work will expand the datasets and enhance inverse kinematics and path-planning strategies to mitigate singularities.

Together, these approaches form a comprehensive framework for trajectory planning in collaborative robotics. By harmonizing human-centered design with computational rigor and safety-critical control, future cobots will be capable of executing complex tasks with grace, speed, and reliability-ushering in a new era of intelligent, human-aware automation.

Chapter 11

Publications of the PhD candidate

Lozer, F. , Scalera, L., Gasparetto, A., and Brandstötter, M. (2025). Neural Network-Based Trajectory Planning of Human-Like Movements in Collaborative Robotics. Submitted to IEEE International Conference on Robotics and Automation (ICRA 2026), Vienna, Austria.

Lozer, F. , Scalera, L., Gasparetto, A., and Brandstötter, M. (2025). Achieving Human-Like Movements with Neural Network-Based Planner in Collaborative Robotics. Extended abstract presented at the 7^{th} edition of the Conference on Robotics and Intelligent Machines (I-RIM 3D 2025), Rome, Italy, 17-19 October 2025.

Lozer, F. , Scalera, L., Boscariol, P., and Gasparetto, A. (2025). Planning optimal minimum-jerk trajectories for redundant robots. Robotics and Autonomous Systems, 192, 105049.

Scalera, L., Lozer, F. , Giusti, A., Gasparetto, A. (2024). An experimental evaluation of robot-stopping approaches for improving fluency in collaborative robotics. Robotica, 42(5), 1386-1402.

Scalera, L., Lozer, F. , Geerinck, J., Breda, A., Totis, F., Polo, F., Giusti, A., Gasparetto, A. (2024). A collaborative robotics application for the assembly of car rear lamps. In Latest Advancements In Mechanical Engineering, Proceedings of the 3rd International Symposium on Industrial Engineering and Automation (ISIEA 2024), 19th-21st June 2024, Bozen-Bolzano, Italy.

Lozer, F. , Scalera, L., Boscariol, P., Gasparetto, A. (2023). An Experimental Setup to Test Time-Jerk Optimal Trajectories for Robotic Manipulators. In International Conference on Robotics in Alpe-Adria Danube Region (pp. 309-316). Cham: Springer Nature Switzerland.

References

Abu-Dakka, F. J., Assad, I. F., Alkhdour, R. M., and Abderahim, M. (2017). Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *The international journal of advanced manufacturing technology*, 89:389–406.

Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

Andronas, D., Kampourakis, E., Papadopoulos, G., Bakopoulou, K., Kotsaris, P. S., Michalos, G., and Makris, S. (2023). Towards seamless collaboration of humans and high-payload robots: an automotive case study. *Robotics and Computer-Integrated Manufacturing*, 83:102544.

Asali, E. and Doshi, P. (2024). Visual IRL for Human-Like Robotic Manipulation. *arXiv preprint arXiv:2412.11360*.

Baressi Šegota, S., Andjelić, N., Lorencin, I., Saga, M., and Car, Z. (2020). Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms. *International journal of advanced robotic systems*, 17(2):1729881420908076.

Barre, P.-J., Bearee, R., Borne, P., and Dumetz, E. (2005). Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. *Journal of Intelligent and Robotic Systems*, 42:275–293.

Biagiotti, L. and Melchiorri, C. (2008). *Trajectory planning for automatic machines and robots*. Springer Science & Business Media.

Boscariol, P., Caracciolo, R., Richiedei, D., and Trevisani, A. (2020). Energy optimization of functionally redundant robots through motion design. *Applied Sciences*, 10(9):3022.

Boscariol, P., Clochiatti, E., Scalera, L., and Gasparetto, A. (2023). A framework for improving the energy efficiency and sustainability of collaborative robots. In *International Workshop IFToMM for Sustainable Development Goals*, pages 47–54. Springer.

Boscariol, P., Gasparetto, A., and Vidoni, R. (2012). Planning continuous-jerk trajectories for industrial manipulators. In *Engineering systems design and analysis*, volume 44861, pages 127–136. American Society of Mechanical Engineers.

Bottin, M. and Rosati, G. (2019). Trajectory optimization of a redundant serial robot using cartesian via points and kinematic decoupling. *Robotics*, 8(4):101.

Braglia, G., Tagliavini, M., Pini, F., and Biagiotti, L. (2023). Online motion planning for safe human–robot cooperation using b-splines and hidden markov models. *Robotics*, 12(4):118.

Byner, C., Matthias, B., and Ding, H. (2019). Dynamic speed and separation monitoring for collaborative robot applications–concepts and performance. *Robotics and Computer-Integrated Manufacturing*, 58:239–252.

Carabin, G. and Scalera, L. (2020). On the trajectory planning for energy efficiency in industrial robotic systems. *Robotics*, 9(4):89.

Chang, Q., Wang, H., Wang, D., Zhang, H., Li, K., and Yu, B. (2021). Motion planning for vibration reduction of a railway bridge maintenance robot with a redundant manipulator. *Electronics*, 10(22):2793.

Chen, D., Li, S., Li, W., and Wu, Q. (2019). A multi-level simultaneous minimization scheme applied to jerk-bounded redundant robot manipulators. *IEEE Transactions on Automation Science and Engineering*, 17(1):463–474.

Chettibi, T., Lehtihet, H., Haddad, M., and Hanchi, S. (2004). Minimum cost trajectory planning for industrial robots. *European Journal of Mechanics-A/Solids*, 23(4):703–715.

Choi, S. H., Park, K.-B., Roh, D. H., Lee, J. Y., Mohammed, M., Ghasemi, Y., and Jeong, H. (2022). An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. *Robotics and Computer-Integrated Manufacturing*, 73:102258.

Cook, C. and Ho, C. (1984). The application of spline functions to trajectory generation for computer-controlled manipulators. In *Computing techniques for robots*, pages 101–110.

Corke, P. and Haviland, J. (2021). Not your grandmother's toolbox–the Robotics Toolbox reinvented for Python. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11357–11363. IEEE.

Dai, C., Lefebvre, S., Yu, K.-M., Geraedts, J. M., and Wang, C. C. (2020). Planning jerk-optimized trajectory with discrete time constraints for redundant robots. *IEEE Transactions on Automation Science and Engineering*, 17(4):1711–1724.

Dragan, A. D., Bauman, S., Forlizzi, J., and Srinivasa, S. S. (2015). Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 51–58.

Ericson, C. (2004). *Real-time collision detection*. CRC Press.

Fabris, G., Scalera, L., Boscariol, P., and Gasparetto, A. (2024a). Experimental analysis and comparison of friction models applied to the UR5e robot. In *6th IFToMM International Symposium on Mechanism Design for Robotics, MEDER 2024. Mechanisms and Machine Science*, volume 166 MMS, page 125 – 133. Springer.

Fabris, G., Scalera, L., and Gasparetto, A. (2024b). Dynamic modelling and energy-efficiency optimization in a 3-DOF parallel robot. *The International Journal of Advanced Manufacturing Technology*, 132(5):2677–2699.

Fang, Y., Hu, J., Liu, W., Shao, Q., Qi, J., and Peng, Y. (2019). Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mechanism and Machine Theory*, 137:127–153.

Fang, Y., Qi, J., Hu, J., Wang, W., and Peng, Y. (2020). An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints. *Mechanism and Machine Theory*, 153:103957.

Faroni, M., Beschi, M., and Pedrocchi, N. (2022). Safety-aware time-optimal motion planning with uncertain human state estimation. *IEEE Robotics and Automation Letters*, 7(4):12219–12226.

Ferraguti, F., Landi, C. T., Singletary, A., Lin, H.-C., Ames, A., Secchi, C., and Bonfè, M. (2022). Safety and efficiency in robotics: the control barrier functions approach. *IEEE Robotics & Automation Magazine*, 29(3):139–151.

Flowers, J. and Wiens, G. (2024). Prediction of human reaching pose sequences in human–robot collaboration. *Journal of Mechanisms and Robotics*, 16(11).

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Medina-Carnicer, R. (2016). Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491.

Gašpar, T., Deniša, M., Radanovič, P., Ridge, B., Savarimuthu, T. R., Kramberger, A., Priggemeyer, M., Roßmann, J., Wörgötter, F., Ivanovska, T., et al. (2020). Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. *Robotics and Computer-Integrated Manufacturing*, 66:101979.

Gasparetto, A., Lanzutti, A., Vidoni, R., and Zanotto, V. (2011). Validation of minimum time-jerk algorithms for trajectory planning of industrial robots. *Journal of Mechanisms and Robotics*, 3(3):031003.

Gasparetto, A. and Zanotto, V. (2007). A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, 42(4):455–471.

Gasparetto, A. and Zanotto, V. (2008). A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing*, 24(3):415–426.

Gaz, C., Cognetti, M., Oliva, A., Giordano, P. R., and De Luca, A. (2019). Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154.

Giusti, A. and Althoff, M. (2017). Efficient computation of interval-arithmetic-based robust controllers for rigid robots. In *2017 First IEEE International Conference on Robotic Computing (IRC)*, pages 129–135. IEEE.

Giusti, A. and Nainer, C. (2022). Inverse Uncertain-Dynamics of Robot Manipulators Using Interval Arithmetic. In *The International Conference of IFToMM ITALY*, pages 661–668. Springer.

GmbH, F. E. (2017). Robot and Interface Specifications. https://frankaemika.github.io/docs/control_parameters.html. [Online; accessed 16-February-2023].

Grushko, S., Vysockỳ, A., Oščádal, P., Vocetka, M., Novák, P., and Bobovskỳ, Z. (2021). Improved mutual understanding for human-robot collaboration: Combining human-aware motion planning with haptic feedback devices for communicating planned trajectory. *Sensors*, 21(11):3673.

Gulletta, G., Erlhagen, W., and Bicho, E. (2020). Human-like arm motion generation: A review. *Robotics*, 9(4):102.

Haddadin, S., Parusel, S., Johannsmeier, L., Golz, S., Gabl, S., Walch, F., Sabaghian, M., Jähne, C., Hausperger, L., and Haddadin, S. (2022). The Franka Emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64.

Hanna, A., Larsson, S., Götvall, P.-L., and Bengtsson, K. (2022). Deliberative safety for industrial intelligent human–robot collaboration: Regulatory challenges and solutions for taking the next step towards Industry 4.0. *Robotics and Computer-Integrated Manufacturing*, 78:102386.

Haug, E. J. (2024). Redundant serial manipulator inverse position kinematics and dynamics. *Journal of Mechanisms and Robotics*, 16(8):081008.

He, Y. and Liu, S. (2021). Analytical inverse kinematics for franka emika panda–a geometrical solver for 7-dof manipulators with unconventional design. In *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 194–199. IEEE.

Hoffman, G. (2019). Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218.

Hoffman, G. and Breazeal, C. (2007). Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics*, 23(5):952–961.

Huang, J., Hu, P., Wu, K., and Zeng, M. (2018). Optimal time-jerk trajectory planning for industrial robots. *Mechanism and Machine Theory*, 121:530–544.

Huang, S., Gao, M., Liu, L., Chen, J., and Zhang, J. (2022). Collision detection for cobots: A back-input compensation approach. *IEEE/ASME Transactions on Mechatronics*, 27(6):4951–4962.

Huang, W., Wang, C., Li, Y., Zhang, R., and Fei-Fei, L. (2024). Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*.

Javaid, M., Haleem, A., Singh, R. P., Suman, R., and Gonzalez, E. S. (2022). Understanding the adoption of Industry 4.0 technologies in improving environmental sustainability. *Sustainable Operations and Computers*, 3:203–217.

Johannessen, L. M. G., Arbo, M. H., and Gravdahl, J. T. (2019). Robot dynamics with URDF & Casadi. In *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 1–6. IEEE.

Kim, E., Yamada, Y., Okamoto, S., Sennin, M., and Kito, H. (2021). Considerations of potential runaway motion and physical interaction for speed and separation monitoring. *Robotics and Computer-Integrated Manufacturing*, 67:102034.

Kim, M., Han, D.-K., Park, J.-H., and Kim, J.-S. (2020). Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay. *Applied Sciences*, 10(2):575.

Kot, T., Wierbica, R., Oščádal, P., Spurnỳ, T., and Bobovskỳ, Z. (2022). Using elastic bands for collision avoidance in collaborative robotics. *IEEE Access*, 10:106972–106987.

Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.

Lacevic, B., Zanchettin, A. M., and Rocco, P. (2022). Safe human-robot collaboration via collision checking and explicit representation of danger zones. *IEEE Transactions on Automation Science and Engineering*, 20(2):846–861.

Lagomarsino, M., Lorenzini, M., De Momi, E., and Ajoudani, A. (2022). Robot trajectory adaptation to optimise the trade-off between human cognitive ergonomics and workplace productivity in collaborative tasks. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 663–669. IEEE.

Li, W., Hu, Y., Zhou, Y., and Pham, D. T. (2023a). Safe human–robot collaboration for industrial settings: a survey. *Journal of Intelligent Manufacturing*, pages 1–27.

Li, Y., Zhang, Y., Razmjoo, A., and Calinon, S. (2023b). Learning robot geometry as distance fields: Applications to whole-body manipulation. *arXiv preprint arXiv:2307.00533*.

Lippi, M. and Marino, A. (2020). Human multi-robot safe interaction: A trajectory scaling approach based on safety assessment. *IEEE Transactions on Control Systems Technology*, 29(4):1565–1580.

Liu, B., Rocco, P., Zanchettin, A. M., Zhao, F., Jiang, G., and Mei, X. (2024a). A real-time hierarchical control method for safe human–robot coexistence. *Robotics and Computer-Integrated Manufacturing*, 86:102666.

Liu, H., Qu, D., Xu, F., Du, Z., Jia, K., Song, J., and Liu, M. (2022). Real-time and efficient collision avoidance planning approach for safe human-robot interaction. *Journal of Intelligent & Robotic Systems*, 105(4):93.

Liu, R. and Liu, C. (2020). Human motion prediction using adaptable recurrent neural networks and inverse kinematics. *IEEE Control Systems Letters*, 5(5):1651–1656.

Liu, X., Wan, B., Liu, Y., Wang, Y., and Zhao, Y. (2024b). Singularity analysis of actuation coordination and new indices for optimal design of redundantly actuated parallel manipulators. *Journal of Mechanisms and Robotics*, 16(10).

Lozer, F., Scalera, L., Boscariol, P., and Gasparetto, A. (2023). An experimental setup to test time-jerk optimal trajectories for robotic manipulators. In *32nd International Conference on Robotics in Alpe-Adria-Danube Region, RAAD 2023. Mechanisms and Machine Science*, volume 135 MMS, page 309 – 316. Springer.

Lozer, F., Scalera, L., Boscariol, P., and Gasparetto, A. (2025). Planning optimal minimum-jerk trajectories for redundant robots. *Robotics and Autonomous Systems*, page 105049.

Lu, J., Zou, T., and Jiang, X. (2022a). A neural network based approach to inverse kinematics problem for general six-axis robots. *Sensors*, 22(22):8909.

Lu, L., Zhang, L., Fan, C., and Wang, H. (2022b). High-order joint-smooth trajectory planning method considering tool-orientation constraints and singularity avoidance for robot surface machining. *Journal of Manufacturing Processes*, 80:789–804.

Lu, S., Ding, B., and Li, Y. (2020). Minimum-jerk trajectory planning pertaining to a translational 3-degree-of-freedom parallel manipulator through piecewise quintic polynomials interpolation. *Advances in Mechanical Engineering*, 12(3):1687814020913667.

Lu, S., Zhao, J., Jiang, L., and Liu, H. (2017a). Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization. *Mathematical Problems in Engineering*, 2017(1):1921479.

Lu, S., Zhao, J., Jiang, L., and Liu, H. (2017b). Time-jerk optimal trajectory planning of a 7-DOF redundant robot. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(5):4211–4222.

Lyu, J., Ruppel, P., Hendrich, N., Li, S., Görner, M., and Zhang, J. (2022). Efficient and collision-free human-robot collaboration based on intention and trajectory prediction. *IEEE Transactions on Cognitive and Developmental Systems*.

Malhan, R. K., Thakar, S., Kabir, A. M., Rajendran, P., Bhatt, P. M., and Gupta, S. K. (2022). Generation of configuration space trajectories over semi-constrained cartesian paths for robotic manipulators. *IEEE Transactions on Automation Science and Engineering*, 20(1):193–205.

Merckaert, K., Convens, B., Nicotra, M. M., and Vanderborght, B. (2024). Real-time constraint-based planning and control of robotic manipulators for safe human–robot collaboration. *Robotics and Computer-Integrated Manufacturing*, 87:102711.

Merckaert, K., Convens, B., Wu, C.-j., Roncone, A., Nicotra, M. M., and Vanderborght, B. (2022). Real-time motion control of robotic manipulators for safe human–robot coexistence. *Robotics and Computer-Integrated Manufacturing*, 73:102223.

Nemec, B., Mavsar, M., Simonič, M., Hrovat, M. M., Škrabar, J., and Ude, A. (2022). Integration of a reconfigurable robotic workcell for assembly operations in automotive industry. In *2022 IEEE/SICE International Symposium on System Integration (SII)*, pages 778–783. IEEE.

Neythalath, N., Brandstötter, M., and Hofbaur, M. (2016). *Redundancy Resolution of a 9 DOF Serial Manipulator Under Hard Task Constraints*, volume 569, pages 31–38.

Ni, J., Mei, J., Ding, Y., Yu, D., Duan, Y., and Le, Y. (2023). A trajectory planning approach for Delta robots considering both motion smoothness and dynamic stress. *Journal of Mechanisms and Robotics*, 15(4):041012.

Paing, M. S. and Uchiyama, N. (2022). A spline-based approach to smooth and time-optimal trajectory generation for cnc machines with guaranteed kinematic constraints. *The International Journal of Advanced Manufacturing Technology*, 121(5):3385–3398.

Palleschi, A., Garabini, M., Caporale, D., and Pallottino, L. (2019). Time-optimal path tracking for jerk controlled robots. *IEEE Robotics and Automation Letters*, 4(4):3932–3939.

Pereira, A., Baumann, M., Gerstner, J., and Althoff, M. (2022). Improving efficiency of human-robot coexistence while guaranteeing safety: Theory and user study. *IEEE Transactions on Automation Science and Engineering*, 20(4):2706–2719.

Pheasant, S. and Haslegrave, C. M. (2018). *Bodyspace: Anthropometry, ergonomics and the design of work*.

Piazzi, A. and Visioli, A. (1998). Global minimum-time trajectory planning of mechanical manipulators using interval analysis. *International Journal of Control*, 71(4):631–652.

Piazzi, A. and Visioli, A. (2000). Global minimum-jerk trajectory planning of robot manipulators. *IEEE Transactions on Industrial Electronics*, 47(1):140–149.

Prianto, E., Kim, M., Park, J.-H., Bae, J.-H., and Kim, J.-S. (2020). Path planning for multi-arm manipulators using deep reinforcement learning: Soft actor–critic with hindsight experience replay. *Sensors*, 20(20):5911.

Qie, X., Kang, C., Zong, G., and Chen, S. (2022). Trajectory planning and simulation study of redundant robotic arm for upper limb rehabilitation based on back propagation neural network and genetic algorithm. *Sensors*, 22(11):4071.

Quiñones, D. P., Paterna, M., De Benedictis, C., Maffiodo, D., Franco, W., and Ferraresi, C. (2023). Contact force regulation in physical human-machine interaction based on model predictive control. *Robotica*, 41(11):3409–3425.

Reiter, A., Müller, A., and Gattringer, H. (2016). Inverse kinematics in minimum-time trajectory planning for kinematically redundant manipulators. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6873–6878. IEEE.

Romero-Ramirez, F. J., Muñoz-Salinas, R., and Medina-Carnicer, R. (2018). Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47.

Safeea, M., Neto, P., and Bearee, R. (2019). On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. *Robotics and Autonomous Systems*, 119:278–288.

Scalera, L., Giusti, A., Vidoni, R., Di Cosmo, V., Matt, D., and Riedl, M. (2020). Application of dynamically scaled safety zones based on the ISO/TS 15066: 2016 for collaborative robotics. *International Journal of Mechanics and Control*, 21(1):41–49.

Scalera, L., Giusti, A., Vidoni, R., and Gasparetto, A. (2022). Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones. *The International Journal of Advanced Manufacturing Technology*, 121(9):6783–6798.

Scalera, L., Lozer, F., Giusti, A., and Gasparetto, A. (2024a). An experimental evaluation of robot-stopping approaches for improving fluency in collaborative robotics. *Robotica*, 42(5):1386–1402.

Scalera, L., Nainer, C., Giusti, A., and Gasparetto, A. (2023). Robust safety zones for manipulators with uncertain dynamics in collaborative robotics. *International Journal of Computer Integrated Manufacturing*, pages 1–13.

Scalera, L., Nainer, C., Giusti, A., and Gasparetto, A. (2024b). Robust safety zones for manipulators with uncertain dynamics in collaborative robotics. *International Journal of Computer Integrated Manufacturing*, 37(7):887–899.

Scalera, L., Vidoni, R., and Giusti, A. (2021). Optimal scaling of dynamic safety zones for collaborative robotics. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3822–3828. IEEE.

Secil, S. and Ozkan, M. (2022). Minimum distance calculation using skeletal tracking for safe human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 73:102253.

Seriani, S., Gallina, P., Scalera, L., and Lughi, V. (2018). Development of n-DoF preloaded structures for impact mitigation in cobots. *Journal of Mechanisms and Robotics*, 10(5):051009.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). Robotics: Modelling, Planning and Control. *Advanced Textbooks in Control and Signal Processing*.

Solak, G. and Ajoudani, A. (2023). Online learning and suppression of vibration in collaborative robots with power tools. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12085–12091. IEEE.

Spitzhirn, M., Benter, M., Heindl, C., Scheder, N., Reisinger, G., Strohmeier, F., and Behrendt, W. (2022). Hybrid work systems—platform-based work planning—designing productive and human-centered work processes. *Zeitschrift für Arbeitswissenschaft*, 76(4):489–509.

Standard, I. (2016). ISO/TS 15066: 2016: Robots and Robotic Devices–Collaborative Robots. *International Organization for Standardization: Geneva, Switzerland*.

Student (1908). The probable error of a mean. *Biometrika*, 6(1):1–25.

Svarny, P., Tesar, M., Behrens, J. K., and Hoffmann, M. (2019). Safe physical HRI: Toward a unified treatment of speed and separation monitoring together with power and force limiting. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7580–7587. IEEE.

Tittel, S. (2021). Analytical solution for the inverse kinematics problem of the franka emika panda seven-dof light-weight robot arm. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 1042–1047. IEEE.

Trigatti, G., Boscariol, P., Scalera, L., Pillan, D., and Gasparetto, A. (2018). A look-ahead trajectory planning algorithm for spray painting robots with non-spherical wrists. In *IFToMM Symp. on Mech. Design for Robotics*, pages 235–242. Springer.

Vidussi, F., Boscariol, P., Scalera, L., and Gasparetto, A. (2021). Local and trajectory-based indexes for task-related energetic performance optimization of robotic manipulators. *Journal of Mechanisms and Robotics*, 13(2):021018.

Wang, L., Wu, Q., Lin, F., Li, S., and Chen, D. (2019). A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators. *IEEE access*, 7:154331–154345.

Wang, Y., Wei, L., Du, K., Liu, G., Yang, Q., Wei, Y., and Fang, Q. (2023). An online collision-free trajectory generation algorithm for human–robot collaboration. *Robotics and Computer-Integrated Manufacturing*, 80:102475.

Wu, G. and Zhang, S. (2022). Real-time jerk-minimization trajectory planning of robotic arm based on polynomial curve optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(21):10852–10864.

Wu, G., Zhao, W., and Zhang, X. (2021). Optimum time-energy-jerk trajectory planning for serial robotic manipulators by reparameterized quintic nurbs curves. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(19):4382–4393.

Yang, B., Xie, S., Chen, G., Ding, Z., and Wang, Z. (2022). Dynamic speed and separation monitoring based on scene semantic information. *Journal of Intelligent & Robotic Systems*, 106(2):35.

Yonetani, R., Taniai, T., Barekatain, M., Nishimura, M., and Kanezaki, A. (2021). Path planning using neural a* search. In *International conference on machine learning*, pages 12029–12039. PMLR.

Zanotto, V., Gasparetto, A., Lanzutti, A., Boscariol, P., and Vidoni, R. (2011). Experimental validation of minimum time-jerk algorithms for industrial robots. *Journal of Intelligent & Robotic Systems*, 64(2):197–219.

Appendix A

Minimum-jerk trajectory planning

A.1 Computation of the Jacobian and its time derivatives

In the following, we briefly recall the expressions for the computation of the i-th column of the Jacobian matrix J, as well as of its first $\dot{J}$ and second time derivatives $\ddot{J}$, according to the Denavit-Hartenberg (DH) convention and with reference to the notation of the classic robotics textbook by Siciliano et al. Siciliano et al. (2009). For the sake of completeness, the cases of both prismatic (p) and revolute (r) joints are reported. According to the DH convention as formulated by Siciliano et al., the axis i denotes the axis of the joint connecting link $i-1$ to link i. The unit vector of the z_i axis is defined along the axis of joint $i+1$: it is the axis about which there is rotation in the case of a revolute joint, or translation in the case of a prismatic joint. In the following, $\mathbf{p}$ is the position vector of the robot end-effector, $\mathbf{p}_{i-1}$ represents the position vector of the $(i-1)$-th reference frame, $\mathbf{z}_{i-1}$ the unit vector of the z axis of the $(i-1)$-th reference frame, whereas ω_{i-1} is the contribution to the angular velocity of the $(i-1)$-th joint. The $(i-1)$-th column of the Jacobian for a prismatic and a revolute joint, respectively, is equal to:

$$J_i, p = \begin{bmatrix} \mathbf{z}_{i-1} \\ \mathbf{0} \end{bmatrix} \tag{A.1}$$

$$J_i, r = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p} - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \tag{A.2}$$

The first time derivative of (A.1) and (A.2) is given by:

$$\dot{J}_i, p = \begin{bmatrix} \omega_{i-1} \times \mathbf{z}_{i-1} \\ \mathbf{0} \end{bmatrix} \tag{A.3}$$

with

$$\omega_{i-1} = \omega_{i-2} \tag{A.4}$$

$$\dot{J}_i, r = \begin{bmatrix} (\omega_{i-1} \times \mathbf{z}_{i-1}) \times (\mathbf{p} - \mathbf{p}_{i-1}) + \mathbf{z}_{i-1} \times (\dot{\mathbf{p}} - \dot{\mathbf{p}}_{i-1}) \\ \omega_{i-1} \times \mathbf{z}_{i-1} \end{bmatrix} \tag{A.5}$$

with

$$\begin{cases} \omega_{i-1} = \mathbf{z}_{i-1} \cdot \dot{\mathbf{q}}_{i-1} + \omega_{i-2} \\ \dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \\ \dot{\mathbf{p}}_{i-1} = \omega_{i-1} \times (\mathbf{p}_{i-1} - \mathbf{p}_{i-2}) + \dot{\mathbf{p}}_{i-2} \end{cases} \tag{A.6}$$

Finally, the $(i-1)$-th column of the second time derivative of the Jacobian for a prismatic joint can be computed as:

$$\ddot{J}_i, p = \begin{bmatrix} \dot{\omega}_{i-1} \times \mathbf{z}_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times \mathbf{z}_{i-1}) \\ \mathbf{0} \end{bmatrix} \tag{A.7}$$

with

$$\begin{cases} \omega_{i-1} = \omega_{i-2} \\ \dot{\omega}_{i-1} = \dot{\omega}_{i-2} \end{cases} \tag{A.8}$$

or, for a revolute joint, as:

$$\ddot{J}_i, r = \begin{bmatrix} (\dot{\omega}_{i-1} \times \mathbf{z}_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times \mathbf{z}_{i-1})) \times (\mathbf{p} - \mathbf{p}_{i-1}) + ... \\ +(\omega_{i-1} \times \mathbf{z}_{i-1}) \times (\dot{\mathbf{p}} - \dot{\mathbf{p}}_{i-1})) + ... \\ +(\omega_{i-1} \times \mathbf{z}_{i-1}) \times (\dot{\mathbf{p}} - \dot{\mathbf{p}}_{i-1}) + \mathbf{z}_{i-1} \times (\ddot{\mathbf{p}} - \ddot{\mathbf{p}}_{i-1}) \\ \dot{\omega}_{i-1} \times \mathbf{z}_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times \mathbf{z}_{i-1}) \end{bmatrix} \tag{A.9}$$

with

$$\begin{cases} \omega_{i-1} = \mathbf{z}_{i-1} \cdot \dot{\mathbf{q}}_{i-1} + \omega_{i-2} \\ \dot{\omega}_{i-1} = (\omega_{i-1} \times \mathbf{z}_{i-1}) \cdot \dot{\mathbf{q}}_{i-1} + \mathbf{z}_{i-1} \cdot \ddot{\mathbf{q}}_{i-1} + \omega_{i-2} \\ \dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \\ \dot{\mathbf{p}}_{i-1} = \omega_{i-1} \times (\mathbf{p}_{i-1} - \mathbf{p}_{i-2}) + \dot{\mathbf{p}}_{i-2} \\ \ddot{\mathbf{p}} = \dot{\mathbf{J}} \cdot \dot{\mathbf{q}} + \mathbf{J} \cdot \ddot{\mathbf{q}} \\ \ddot{\mathbf{p}}_{i-1} = \dot{\omega}_{i-1} \times (\mathbf{p}_{i-1} - \mathbf{p}_{i-2}) + \omega_{i-1} \times (\dot{\mathbf{p}}_{i-1} - \dot{\mathbf{p}}_{i-2}) + \ddot{\mathbf{p}}_{i-2} \end{cases} \tag{A.10}$$

Fundings

I want to thank:

- the PhD program of the University of Genoa;
- the projects of the Mechatronics and Robotics Laboratory (mec-rob-lab.uniud.it) and the AI4HRC (Artificial Intelligence For Human-Robot Collaboration) laboratory of the University of Udine;
- Marelli Automotive Lighting s.p.a, for having hosted me during my traineeship period in factory;
- ADMiRE Research Center, for having hosted me during my traineeship period abroad;
- the iNEST project of PNRR.

Acknowledgements

And I would like to acknowledge my family, my friends, my colleagues and everybody who helped me.