DRIM

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

Administrative Headquarters: Universita di Genova

Probabilistic and Adaptive Methods for Robotic Manipulation and Robot Hand Control under Interaction Uncertainty

by

Alex Pasquali

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

March 2026

Gianluca Palli	Supervisor
Antonio Sgorbissa	Head of the PhD program

Thesis Jury:

Isiah Zaplana	External examiner
Sylvain Calinon	External examiner

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi" - DEI

I would like to dedicate this thesis primarily to my family. If I have reached this point in my career, it is not only thanks to my own efforts, but also to a team effort that may have seemed invisible to others, yet was always clear to me.

I would also like to dedicate it to all the friends and colleagues who stayed close to me, who laughed with me even when there was nothing to laugh about.

I would like to close this dedication with three special mentions:

Ridere, Ridere, Ridere ancora

Andreosky e Pacionky fratelli nel tempo

Dregher il terribile

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.

Alex Pasquali
March 2026

Acknowledgements

I would like to sincerely thank my supervisor, Gianluca Palli, for his guidance and support throughout this PhD journey.

I would also like to thank all the colleagues who helped me in the development of the projects and research activities that led to this thesis and allowed me to bring these three years of doctoral study to a successful conclusion.

Abstract

This dissertation studies robotic manipulation under interaction uncertainty, including uncertain contact, noisy sensing, human input variability, task reconfiguration, and evaluation variability. Its central claim is that performance improves when sensing, adaptive control, human-in-the-loop programming, probabilistic inference, and evaluation are designed as one coupled pipeline rather than as isolated modules.

On the industrial manipulation side, the dissertation presents a stiffness-aware genetic optimization method for elastic interaction, a sensing-integrated dual-arm architecture for wiring harness assembly, and a modular human-in-the-loop design for collaborative robot teaching. The optimization method uses force measurements directly in the fitness loop, avoiding explicit deformable-object modeling in the tested scenarios. The assembly cell demonstrated end-to-end execution over repeated trials, and the connector-programming study showed that users could program the task and later trigger autonomous execution, supporting a proof-of-concept assessment.

On the robotic hand side, the dissertation presents probabilistic shared-autonomy methods for grip-force regulation and extends them to hierarchical force-aware skill encoding. By combining sEMG intent estimation, tactile sensing, and Hidden Markov Models, the proposed controllers improved grip-force controllability in the evaluated tasks. The evidence comes from controlled user studies and a pilot condition with one participant with upper-limb loss, while the programming-by-demonstration extension supports autonomous reproduction of demonstrated force-aware skills under limited object variation.

The dissertation also introduces a local probabilistic nonlinear latent model for reach-to-grasp reconstruction and iHandVR, an immersive platform for prosthetic evaluation. The latent model improved reconstruction quality over the considered global and linear baselines on the studied datasets, and iHandVR combines device-specific simulation, real-time control, and synchronized multimodal acquisition for functional and psychophysical assessment.

Taken together, these results support a narrower thesis: explicit treatment of interaction uncertainty can improve manipulation performance and evaluation quality within the studied tasks, datasets, hardware platforms, and participant groups.

Table of contents

List of figures

List of tables

List of Abbreviations

ANOVA Analysis of Variance

CAD Computer-Aided Design

CE Conformite Europeenne

CPU Central Processing Unit

CV Cross-Validation

DC Direct Current

DLO Deformable Linear Object

DoF Degree of Freedom

DTW Dynamic Time Warping

EM Expectation-Maximization

EMG Electromyography

FEA Finite Element Analysis

GA Genetic Algorithm

GMR Gaussian Mixture Regression

GP Gaussian Process

GP-LVM Gaussian Process Latent Variable Model

H-HMM Hierarchical Hidden Markov Model

HITL Human-In-The-Loop

HMD Head-Mounted Display

HMM Hidden Markov Model

IB Intentional Binding

IL Incremental Learning

IMU Inertial Measurement Unit

LED Light-Emitting Diode

MLP Multi-Layer Perceptron

NMF Non-negative Matrix Factorization

PCA Principal Component Analysis

PbD Programming by Demonstration

PD Proportional-Derivative

PWM Pulse Width Modulation

RBF Radial Basis Function

RI Risk of Instability

RMS Root Mean Square

RMSE Root Mean Square Error

ROS Robot Operating System

ROS-TCP Robot Operating System - Transmission Control Protocol

sEMG Surface Electromyography

SE(3) Special Euclidean Group in 3D

SHAP Southampton Hand Assessment Procedure

ToF Time-of-Flight

UI User Interface

URDF Unified Robot Description Format

VR Virtual Reality

XML Extensible Markup Language

Chapter 1

Introduction

This chapter defines the scientific scope of the dissertation, motivates the addressed research problems, and clarifies the original contributions. The perspective is intentionally broad at the beginning and progressively narrows toward interaction-aware sensing, modular human-in-the-loop programming, and probabilistic and adaptive methods for robotic manipulation and robotic hand control, together with the supporting system and evaluation choices that structure the thesis.

1.1 Domain Overview

Robotic manipulation is a strategic capability for industrial automation, assistive robotics, and upper-limb prosthetics. Across these domains, the central difficulty is not motion generation alone, but the need to act effectively when object behavior, contact conditions, sensing quality, and user input are only partially known. In manufacturing, many high-value tasks still rely on human dexterity because Deformable Linear Objects (DLOs) remain difficult to manipulate with stable quality and repeatability (Nguyen et al. (2020); Olbrich and Lackinger (2022); Trommnau et al. (2019); Zhu et al. (2022)). In human-centered applications, similar uncertainty appears in grip-force regulation and shared control, where user intent and interaction conditions must be interpreted online (Thomas et al. (2023); Wen et al. (2020); Zhuang et al. (2019)).

Recent progress in tactile sensing, wearable sEMG interfaces, and learning-based methods has expanded the practical capabilities of robotic systems (Cirillo et al. (2021); Farina et al. (2014); Meattini et al. (2018); Romeo et al. (2021)). These advances show that interaction uncertainty can be observed, estimated, and partially compensated rather than merely tolerated. At the same time, they also reveal a persistent fragmentation: adaptive

control, probabilistic inference, sensing, and validation are often developed as separate improvements instead of being organized as parts of one coherent manipulation pipeline.

This fragmentation is especially limiting in prosthetic-oriented and contact-rich scenarios, where functional assessment, user feedback, and reproducible evaluation all matter when comparing systems (Atashzar et al. (2021); Kyberd et al. (2009); Piggott et al. (2016)). It is also limiting in collaborative industrial settings, where intuitive robot programming and frequent reconfiguration are often treated as interface problems detached from the sensing and control architecture. As a result, the broader transition that remains incomplete in current practice is not simply from "avoiding contact" to "using contact," but from isolated local solutions to integrated manipulation systems that can operate under uncertain interaction.

1.2 Scientific Problem and Literature Gap

The core scientific problem addressed in this dissertation is the following: effective robotic manipulation remains difficult when interaction uncertainty spans the physical task, the sensing process, the control interface, and the evaluation setting.

The first gap concerns uncertain interaction in manipulation tasks. In path planning and optimization, heuristic and local methods are widely used and often effective for specific problems, but they generally require problem-dependent design choices when constraints and interactions become more complex (Boussaïd et al. (2013); Hwang et al. (2003); Mac et al. (2016)). In DLO manipulation, several works propose dedicated models or task-specific pipelines (Huo et al. (2022); Lv et al. (2022); Yu et al. (2023)), yet comparatively few studies report integrated systems that jointly address industrial wiring tasks, sensing, and manipulation execution within one workflow (Galassi et al. (2022); Monguzzi et al. (2023); Trommnau et al. (2019)).

The second gap concerns the limits of isolated sensing and control schemes. Model-based grasping and force-prediction methods can be effective for deformable or fragile objects, but they are typically developed under task- and model-specific assumptions (Wen et al. (2020); Zaidi et al. (2017)). Although tactile feedback and shared-control strategies have shown measurable benefits, existing studies are usually validated on specific tasks and do not by themselves solve continuous grip-force regulation under uncertainty as a general problem (Selvaggio et al. (2019); Thomas et al. (2023); Zhuang et al. (2019)). This indicates the need for adaptive control mechanisms that can explicitly use uncertain interaction measurements during execution rather than relying only on fixed mappings or offline assumptions.

The third gap concerns human-in-the-loop programming and interaction architecture. Collaborative robots increasingly rely on multimodal interfaces, shared-autonomy strategies, and wearable sensing, yet many proposed systems remain monolithic or task-specific, making reuse, verification, and reconfiguration difficult in realistic industrial settings. This motivates modular interaction architectures in which intent decoding, feedback delivery, supervisory orchestration, and low-level execution are explicitly separated while remaining functionally integrated.

The fourth gap concerns representation and inference under temporal variability. Low-dimensional grasp abstractions are useful, but prior work also shows that temporal context, nonlinear structure, and higher-order components matter when grasping motions are phase-dependent (Romero et al. (2010); Santello et al. (1998); Xu et al. (2016)). This motivates probabilistic phase inference and probabilistic latent representations in those parts of the dissertation where sequential uncertainty and trajectory structure are central.

The fifth gap concerns reproducible evaluation. In prosthetic-oriented research, standardized functional assessment, realistic interaction modeling, and accessible rehabilitation or testing environments are all important, but they are often addressed separately rather than within one auditable framework (Atashzar et al. (2021); Bruni et al. (2022); Kyberd et al. (2009); Piggott et al. (2016)). Without such integration, it is difficult to compare control strategies across physical and virtual settings or to relate functional outcomes to underlying interaction variables.

This dissertation addresses these gaps through a methodological view of robotic manipulation centered on interaction-aware sensing, modular human-in-the-loop programming, probabilistic inference, adaptive control, and reproducible evaluation. Within this view, contact is one important manifestation of uncertain interaction, sensing is a means of observing that uncertainty, system integration enables deployment in realistic settings, task programming is treated as part of the control architecture rather than as an external interface layer, and reproducible evaluation is part of the methodological answer rather than a final afterthought.

1.3 Motivation of the Work

The motivation of this dissertation is fivefold.

From a sensing standpoint, effective manipulation requires access to interaction-relevant information rather than kinematics alone. In both industrial and prosthetic settings, tactile, force-related, and physiological measurements provide access to variables that are otherwise

hidden or only indirectly observable (Cirillo et al. (2021); Farina et al. (2014); Romeo et al. (2021)). This motivates sensing strategies that transform uncertain interaction into measurable quantities for decision and control.

From a control standpoint, fixed or purely feedforward schemes are often insufficient when object properties, environmental constraints, or human inputs vary during execution. Manipulation performance therefore depends on adaptive mechanisms that can reshape the control action online according to the current interaction state (Mason (2018); Thomas et al. (2023); Wen et al. (2020)). This motivates the development of adaptive control formulations that remain effective across task phases and operating conditions.

From an interaction-design standpoint, collaborative deployment also depends on how robots are programmed, corrected, and reconfigured by users in realistic settings. This motivates modular human-in-the-loop architectures that combine intuitive intent interfaces, feedback channels, and explicit supervisory logic rather than treating robot programming as an isolated software-layer concern.

From an inference standpoint, uncertainty is intrinsic to online intent estimation, phase recognition, and trajectory representation, especially when biological signals are noisy and motion structure is time-varying (Dragan and Srinivasa (2013a); Rabiner (1989); Romero et al. (2010); Starke et al. (2020); Xu et al. (2016)). This motivates the use of probabilistic sequence models and probabilistic latent representations where deterministic descriptions become too rigid or too coarse.

From an evaluation standpoint, scalable and repeatable validation is required to bridge algorithm development and benchmark-quality evidence (Atashzar et al. (2021); Kyberd et al. (2009); Piggott et al. (2016)). This motivates integrated physical-plus-virtual evaluation workflows in which interaction, control, timing, and functional outcomes are jointly observable and comparable.

Therefore, the dissertation adopts a unified research strategy centered on probabilistic and adaptive methods, combined with sensing, modular human-in-the-loop programming, system integration, and reproducible evaluation to address robotic manipulation under interaction uncertainty.

1.4 Objectives and Contributions

The general objective of this dissertation is captured by the following statement: *robotic manipulation and robotic hand control become more effective under interaction uncertainty when sensing, adaptive control, modular human-in-the-loop programming by demonstration,*

probabilistic inference, and reproducible evaluation are designed as a coupled pipeline across physical and virtual environments.

Accordingly, the dissertation is not organized around a single algorithmic family, but around a methodological core that combines probabilistic and adaptive methods with interaction-aware system design: data-driven adaptation is introduced where interaction forces dominate task quality; modular human-in-the-loop programming is introduced where collaborative reconfiguration and task specification are central; probabilistic models are introduced where temporal uncertainty and latent structure are decisive; and system integration and reproducible evaluation are introduced where deployment and comparison across control strategies require consistent physical and virtual protocols.

This thesis makes six integrated contributions.

1. **Claim:** Data-driven adaptive optimization can improve path generation when elastic interaction uncertainty directly affects manipulation quality.
   Evidence: The dissertation develops and experimentally evaluates a stiffness-aware genetic algorithm for elastic-object path optimization in the broader context of DLO manipulation and wiring tasks (Galassi et al. (2022); Pasquali et al. (2024); Trommnau et al. (2019)).

2. **Claim:** Sensing-rich system integration can turn uncertainty-aware execution into a property of the overall manipulation architecture rather than of a single algorithmic module.
   Evidence: The dissertation presents a dual-arm mechatronic architecture for wiring harness assembly that integrates tactile sensing, modular trajectory generation, depth-based inspection, and dedicated tooling into one execution loop (Laudante et al. (2025)).

3. **Claim:** Modular human-in-the-loop programming architectures can make collaborative industrial robot teaching more direct and structurally reusable under reconfiguration uncertainty.
   Evidence: The dissertation develops and validates a multimodal design pattern that separates intent decoding, feedback delivery, supervisory finite-state orchestration, and incremental retraining in a robotic connector assembly case study, enabling end-to-end task programming followed by autonomous execution (Braun et al. (2025); Caporali et al. (2026); Meattini et al. (2018)).

4. **Claim:** Probabilistic phase inference and hierarchical shared-autonomy mechanisms enable controllable grip-force regulation and force-aware skill encoding in human-in-the-loop robotic grasping.
   Evidence: The dissertation develops HMM-based methods that combine sEMG intent estimation, tactile sensing, adaptive gain modulation, and force-aware programming by demonstration for regulation and skill reproduction (Bernardini et al. (2025); Meattini et al. (2018); Pasquali et al. (2026b); Rabiner (1989)).

5. **Claim:** Local probabilistic latent representations outperform global models in phase-dependent reach-to-grasp reconstruction on the studied datasets.
   Evidence: The dissertation evaluates local probabilistic nonlinear latent models against global and linear baselines for reach-to-grasp reconstruction, in a problem setting motivated by prior spatio-temporal and nonlinear synergy studies (Romero et al. (2010); Wang et al. (2026); Xu et al. (2016)).

6. **Claim:** Immersive, device-specific simulation enables reproducible evaluation without decoupling interaction dynamics from functional assessment.
   Evidence: The proposed virtual platform integrates real-time control, multimodal sensing, and functional assessment in VR, while drawing on physics-based simulation and standardized prosthetic evaluation practices (Kyberd et al. (2009); Pasquali et al. (2026a); Piggott et al. (2016); Todorov et al. (2012)).

Together, these six contributions define the methodological profile of the dissertation. At its core, that profile is shaped by probabilistic and adaptive methods, while modular task programming, system integration, and reproducible evaluation ensure applicability and comparability across industrial and prosthetic-oriented settings. Across the chapters, validation is reported through quantitative categories such as force-related metrics, reconstruction error, success rate, completion time, and synchronized functional measurements, with statistically grounded comparisons whenever applicable.

1.5 Dissertation Structure

The dissertation is organized as follows.

- Chapter 2 addresses adaptive data-driven path optimization for robotic manipulation of elastic objects through a stiffness-aware genetic algorithm (Pasquali et al. (2024)).

- Chapter 3 presents a sensing-rich mechatronic integration framework for dual-arm robotic wiring harness assembly, serving as an enabling system chapter for manipulation under interaction uncertainty (Laudante et al. (2025)).
- Chapter 4 introduces a modular human-in-the-loop design pattern for intuitive industrial robot programming via neuromuscular and multimodal interaction, bridging the system-level manipulation architecture and the later shared-autonomy contributions.
- Chapter 5 introduces an HMM-based shared-autonomy method for grip-strength regulation in sEMG-driven robotic hand control (Bernardini et al. (2025)).
- Chapter 6 extends the probabilistic shared-autonomy paradigm toward hierarchical force-aware HITL control and programming of robotic grasping (Pasquali et al. (2026b)).
- Chapter 7 proposes probabilistic local nonlinear latent representations to improve reach-to-grasp trajectory reconstruction in synergy subspaces (Wang et al. (2026)).
- Chapter 8 presents iHandVR, an immersive platform for reproducible functional testing and prosthetic hand control with integrated real-time sensing and evaluation tools (Pasquali et al. (2026a)).
- Chapter 9 provides the concluding synthesis of the dissertation, critically discusses scope and limitations, and outlines future research directions.

Overall, the dissertation follows a coherent path from problem framing to method design, system integration, task programming, and validation in both physical and virtual environments. Each chapter contributes to the same central thesis: effective robotic manipulation under interaction uncertainty requires probabilistic and adaptive methods embedded within a broader framework of sensing, modular human-in-the-loop programming, system integration, and reproducible evaluation across the full manipulation loop.

Chapter 2

Adaptive Path Optimization under Elastic Interaction

2.1 Problem Formulation

Genetic algorithms (GAs) are a family of optimization strategies inspired by natural selection and genetics (Lambora et al. (2019)). They emulate evolutionary processes (Vikhar (2016)) by applying selection, crossover, and mutation to a population of candidate solutions, gradually improving them for the task (Leardi (2007)). A GA typically starts from a random population of individuals and evaluates each one through a fitness measure that expresses problem performance.

Over the last decades, GAs have drawn increasing attention for tackling non-convex and nonlinear optimization problems (Gallagher and Sambridge (1994)). Their evolutionary nature makes them effective for multi-objective (Konak et al. (2006)), constrained (Pasquali et al. (2023)), and adaptive optimization tasks (Srinivas and Patnaik (1994)). Yet, achieving strong performance requires careful choices, including a representation that matches the evolutionary paradigm and well-designed crossover and mutation operators (Boussaïd et al. (2013)).

Trajectory and path optimization is a prominent application area for genetic algorithms (Pasquali et al. (2023); Roberge et al. (2013)), with uses in robotics, logistics, transportation design, and route planning (Galassi and Palli (2021)). In particular, optimal path search in three-dimensional spaces is crucial in many fields, where effective navigation and motion solutions directly affect practical outcomes (Chelouah and Siarry (2000)).

For robotic path finding, several computational approaches have been explored. Traditional heuristic search methods (Mac et al. (2016)) and local optimization techniques (Hwang et al. (2003)) provide partial solutions but are often limited in efficiency.

This chapter develops a stiffness-aware genetic algorithm to efficiently evaluate feasible paths for robotic manipulation of elastic objects. The formulation addresses a dual objective: minimizing the forces perceived by the robot and minimizing the traveled distance. The method uses a decreasing exploration ratio to generate mutated paths at each task cycle, allowing new populations to progressively improve the solution. By leveraging real-time force measurements at each path key point during task execution, the system can improve the solution quality within a predefined iteration budget.

Because the fitness evaluation depends solely on sensor measurements, the approach is model-free. Avoiding an analytical model of flexible objects increases flexibility and generality, given the well-known complexity of modeling and parameter identification for such systems (Caporali et al. (2024); Galassi et al. (2022)).

Unlike traditional GA-based planners that rely mainly on geometric fitness terms or pre-modeled elastic dynamics, the proposed formulation embeds real-time force observables directly into the evolutionary selection process. This shifts the methodological focus from offline model fidelity to online interaction evidence: the quality of each individual is determined by what the robot actually experiences during execution, not by a surrogate deformation model. In this sense, the contribution is not only applicative but methodological, because force observables become first-class variables in mutation pressure, selection, and convergence behavior.

The case study provides proof-of-concept evidence that the algorithm can explore the solution space and identify paths that reduce the applied forces in the tested elastic setups. This is supported by a preliminary analysis of parameter configurations through simulated experiments. After selecting the best parameter set, it is applied in multiple trials on the real robotic setup.

The remainder of this chapter is organized as follows: Section 2.2 presents the mathematical formulation of the proposed GA, including its description, fitness definition, and parameter analysis/selection; Section 2.3 describes the experimental setup and discusses results; Section 2.4 concludes the chapter.

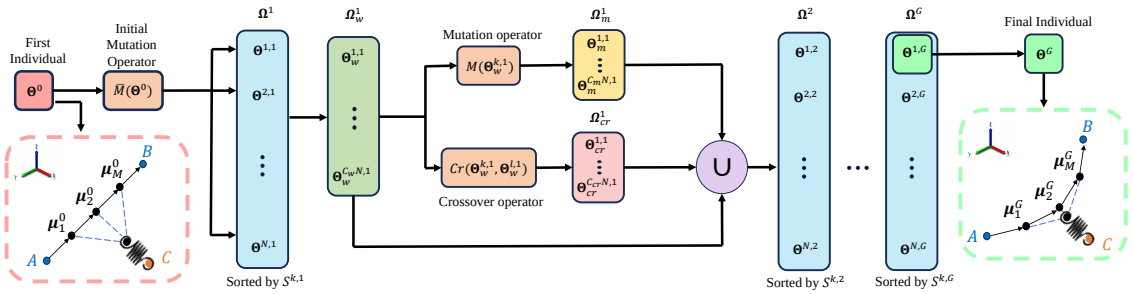


Figure 2.1 Overview of the genetic algorithm, where Θ denotes individuals and Ω denotes generations.

2.2 Force-Aware Evolutionary Framework

This section presents the GA used to solve the path finding optimization problem in robotic systems subject to external elastic forces. Each individual is encoded by a mean position (μ) and a standard deviation (σ), and individuals are grouped into populations across generations (Ω). The algorithm includes mutation, crossover, initialization, and a parameter selection stage that explores combinations to tune performance (Fig. 2.1). Simulated experiments with springs of different constants guide the final choice of parameters.

2.2.1 Scalability and Computational Footprint

The computational cost is dominated by fitness evaluation, since each generation requires evaluating all individuals over all path key points. In qualitative terms, the complexity scales as $\mathcal{O}(G \cdot N \cdot T)$, where G is the number of generations, N is the population size, and T is the number of key points in each path. Mutation and crossover add lower-order operations with the same dependence on N and T. Practically, increasing T improves path resolution but raises evaluation time linearly; increasing N improves exploration but increases per-generation cost; increasing G improves convergence opportunities but extends runtime proportionally. The selected parameter set in this chapter reflects this trade-off and was chosen to remain computationally tractable on real hardware while preserving force-reduction performance.

2.2.2 Individuals, Population and Generations

The genetic encoding in our framework has two components: the mean workspace position (μ) and the standard deviation (σ), which together govern the exploration–exploitation

balance during the evolutionary search. Populations (Ω) collect these genetic parameters across generations. The genetic representation of an individual Θ is structured as follows:

$$\Theta^{k,i} = \{\mu^{k,i}, \sigma^{k,i}\}, \quad k = 1, \ldots, N.$$

Here, $\Theta^{k,i}$ refers to the genetic parameters of individual k in generation i. It contains the mean position $\mu^{k,i}$ in the solution space and the standard deviation $\sigma^{k,i}$, which controls exploration and exploitation during the search. A population Ω is represented as:

$$\Omega^{i} = \{\Theta^{1,i}, \ldots, \Theta^{N,i}\} \quad i = 1, \ldots, G.$$

Ω^{i} denotes the set of all individuals in generation i. The $\mu^{k,i}$ represents the mean position matrix of individual k in generation i. Each row corresponds to one optimized path point, while the columns span the $M = 3$ workspace coordinates (x, y, z).

$$\mu^{k,i} = \begin{bmatrix} x_1^{k,i} & y_1^{k,i} & z_1^{k,i} \\ \vdots & \vdots & \vdots \\ x_T^{k,i} & y_T^{k,i} & z_T^{k,i} \end{bmatrix}, \quad x, y, z \in \mathbb{R}.$$

The parameters N, G, and T represent the population size, the number of generations, and the number of optimized path points, respectively. The workspace dimension is fixed to $M = 3$.

$$N, G, T \in \mathbb{N}^{*}, \quad M = 3.$$

The $\sigma^{k,i}$ denotes the standard deviation (exploration ratio) associated with individual k in generation i, and it must be non-negative.

$$\sigma^{k,i} \in \mathbb{R}, \quad \sigma^{k,i} \geq 0.$$

2.2.3 Fitness Function Definition

The algorithm requires a properly defined fitness function, together with suitable genetic operators, to quantify the quality of candidate solutions. For this reason, our fitness function $S^{k,i}$ incorporates both spatial distribution $\Phi^{k,i}$ and force $\Psi^{k,i}$ terms.

$$S^{k,i} = S^{k,i}(\Phi^{k,i}, \Psi^{k,i}) = \sum_{j=1}^{T} S_j^{k,i} =$$

$$= \sum_{j=1}^{T} ((\Phi_j^{k,i})^2 \cdot \Psi_j^{k,i}) \cdot (1 + |\Phi_j^{k,i} - \Psi_j^{k,i}|).$$

The term $(\Phi_j^{k,i})^2 \cdot \Psi_j^{k,i}$ combines spatial distribution and force exchange at point j, while $(1 + |\Phi_j^{k,i} - \Psi_j^{k,i}|)$ adjusts the contribution using the absolute difference between the spatial and force scores.

$\Psi^{k,i}$ accounts for the forces acting on the robot along the path by computing the magnitude of applied forces relative to a maximum allowable force F_{max}. This component aims to reduce the impact of external forces on the path.

$$\Psi_j^{k,i} = \Psi_j^{k,i}(\mu_j^{k,i}) = 1 + \frac{\sqrt{F_x^2 + F_y^2 + F_z^2}}{F_{max}}, \quad F_{max} > 0.$$

To compute the spatial distribution term $\Phi^{k,i}$, we define points A and B as the start and end points of the path in three-dimensional space.

$$A = [x_A, y_A, z_A], \quad B = [x_B, y_B, z_B].$$

We then consider the matrix $P^{k,i}$, which collects the sequence of points that define the path for individual $\Theta^{k,i}$. The function $d(P_j^{k,i})$ computes the Euclidean distance between two consecutive points along the path.

$$P^{k,i} = \begin{bmatrix} A \\ \mu^{k,i} \\ B \end{bmatrix}, \quad d(P_j^{k,i}) = |P_j^{k,i} - P_{j+1}^{k,i}|.$$

$\Phi^{k,i}$ evaluates the spatial distribution of points by measuring the deviation of inter-point distances from an ideal spacing D. This term penalizes deviations from the desired spacing.

$$D = \frac{|A - B|}{T + 1},$$

$$\Phi_j^{k,i} = \left(1 + \frac{1}{2}|d(P_j^{k,i}) - D| + \frac{1}{2}|d(P_{j+1}^{k,i}) - D|\right)^2.$$

2.2.4 Mutations, Crossovers and Initialization

For mutation, we define the function $M(\Theta^{k,i})$, which generates a new individual $\Theta^{n,i+1}$ from the current individual $\Theta^{k,i}$. The operations required are:

$$\Theta^{n,i+1} = M(\Theta^{k,i}) = \{\mu^{n,i+1}, \sigma^{n,i+1}\},$$

$$\zeta^{k,i} = [\zeta_1^{k,i}, \ldots, \zeta_T^{k,i}], \quad \zeta_t^{k,i} = \frac{S_t^{k,i}}{S^{k,i}},$$

$$m = \arg\max_t (\zeta_t^{k,i} \cdot U(0,1)_t), \quad t = 1, \ldots, T,$$

$$\begin{cases} \mu_t^{n,i+1} = \mu_t^{k,i} & \text{if } t \neq m \\ \mu_t^{n,i+1} = \mu_t^{k,i} + U(-1,1)_{1\times 3} \cdot \sigma^{k,i} & \text{if } t = m, \end{cases}$$

$$t = 1, \ldots, T,$$

$$\sigma^{n,i+1} = \delta \cdot \sigma^{k,i}, \quad \delta \in (0,1].$$

After computing mutation probabilities $\zeta^{k,i}$, which represent the likelihood of mutating each point $\mu_t^{k,i}$, we select the index m. The point $\mu_m^{k,i}$ is mutated by adding a random vector $U(-1,1)_{1\times 3} \cdot \sigma^{k,i}$. Finally, the new individual's standard deviation $\sigma^{n,i+1}$ is updated using the scaling factor δ.

For crossover, we define the operator $Cr(\Theta^{k,i}, \Theta^{l,i})$, which produces a new individual $\Theta^{c,i+1}$ from two parents $\Theta^{k,i}, \Theta^{l,i}$. The procedure is:

$$\Theta^{k,i} = \{\mu^{k,i}, \sigma^{k,i}\}, \quad \Theta^{l,i} = \{\mu^{l,i}, \sigma^{l,i}\},$$

$$\Theta^{c,i+1} = Cr(\Theta^{k,i}, \Theta^{l,i}) = \{\mu^{c,i+1}, \sigma^{c,i+1}\},$$

$$\begin{cases} \mu_t^{c,i+1} = \mu_t^{k,i} & \text{if } \zeta_t^{k,i} \cdot U(0,1) \leq \zeta_t^{l,i} \cdot U(0,1) \\ \mu_t^{c,i+1} = \mu_t^{l,i} & \text{otherwise} \end{cases}$$

$$t = 1, \ldots, T,$$

$$\sigma^{c,i+1} = \delta \cdot \sigma^{k,i} = \delta \cdot \sigma^{l,i}, \quad \delta \in (0,1].$$

The crossover operator chooses points from either $\mu^{k,i}$ or $\mu^{l,i}$ by comparing their mutation probabilities $\zeta_t^{k,i}$ and $\zeta_t^{l,i}$ against random values $U(0,1)$. The standard deviation $\sigma^{c,i+1}$ is then updated with the scaling factor δ.

Initialization proceeds as follows:

$$\Omega^1 = \{\Theta^{1,1}, \ldots, \Theta^{N,1}\},$$

$$\Theta^0 = \{\mu^0, \sigma^0\}.$$

The initial individual Θ^0 is defined by the initial mean vector μ^0 and the initial standard deviation σ^0. The initial mean vector μ^0 is computed with a linear interpolation function $L_I(\ldots)$ between the start point A and the end point B, using T optimized path points. The initial standard deviation σ^0 is chosen as the distance between consecutive points of the corresponding linear path:

$$\mu^0 = L_I(A, B, T), \quad \sigma^0 = \frac{|A - B|}{T + 1}.$$

The scaling factor δ is computed as:

$$\delta = \sqrt[G]{\frac{\sigma^G}{\sigma^0}}, \quad 0 \leq \sigma^G \leq \sigma^0,$$

where σ^G is the desired final value of the standard deviation. Each individual $\Theta^{k,1}$ in the population is initialized using the initial mutation operator $\bar{M}$ applied to Θ^0 (Fig. 2.1).

$$\Theta^{k,1} = \bar{M}(\Theta^0) = \{\mu^{k,1}, \sigma^{k,1}\}, \quad k = 1, \ldots, N.$$

All components of the mean vector $\mu^{k,1}$ are updated by adding a random value from a uniform distribution, scaled by the initial standard deviation σ^0, to μ_t^0.

$$\mu_t^{k,1} = \mu_t^0 + U(-1, 1) \cdot \sigma^0, \quad t = 1, \ldots, T.$$

Finally, the standard deviation $\sigma^{k,1}$ for each individual is updated by the scaling factor δ:

$$\sigma^{k,1} = \delta \cdot \sigma^0.$$

2.2.5 Algorithm Parameters: Description

This part explains how the population is partitioned and manipulated across successive generations of the genetic algorithm. The coefficients C_w, C_m, and C_{cr} represent the proportions of

the population assigned to selection, mutation, and crossover.

$$C_w, C_m, C_{cr} \in [0,1], \quad C_w + C_m + C_{cr} = 1.$$

These coefficients are multiplied by N to obtain the number of individuals in each category:

$$C_w \cdot N, C_m \cdot N, C_{cr} \cdot N \in \mathbb{N} \quad C_w \geq \max(C_m, C_{cr}).$$

$$\Omega^i = \{\Theta^{1,i}, \ldots, \Theta^{N,i}\} \text{ with } S^{1,i} \leq \cdots \leq S^{N,i},$$

the population Ω^i in the i-th generation is sorted according to fitness. The population is then split into three subgroups Ω^i_w, Ω^i_m, and Ω^i_{cr}, which correspond to selected, mutated, and crossed individuals (Fig. 2.1).

$$\Omega^i_w = \{\Theta^{1,i}_w, \ldots, \Theta^{C_w \cdot N,i}_w\},$$

$$\Theta^{k,i}_w = \Theta^{k,i}, \quad k = 1, \ldots, C_w \cdot N,$$

where the individuals $\Theta^{k,i}_w$ in Ω^i_w are preserved unchanged and directly promoted to the next generation.

$$\Omega^i_m = \{\Theta^{1,i}_m, \ldots, \Theta^{C_m \cdot N,i}_m\},$$

$$\Theta^{k,i}_m = M(\Theta^{k,i}_w), \quad k = 1, \ldots, C_m \cdot N,$$

where Ω^i_m collects mutated individuals derived from the selected group.

$$\Omega^i_{cr} = \{\Theta^{1,i}_{cr}, \ldots, \Theta^{C_{cr} \cdot N,i}_{cr}\},$$

$$\Theta^{k,i}_{cr} = Cr(\Theta^{k,i}_w, \Theta^{l,i}_w), \quad k = 1, \ldots, C_{cr} \cdot N,$$

$$l \in [1, C_w \cdot N] \text{ random choice with } l \neq k,$$

where Ω^i_{cr} contains crossed individuals $\Theta^{k,i}_{cr}$. Each new component is produced by crossover between two members of the selected subgroup.

The next generation Ω^{i+1} is then formed by merging the three subgroups:

$$\Omega^{i+1} = \{\Theta^{1,i+1}, \ldots, \Theta^{N,i+1}\} = \Omega^i_w \cup \Omega^i_m \cup \Omega^i_{cr}.$$

Table 2.1 Parameter configurations including number of individuals (N), selection rate (C_w), mutation rate (C_m), crossover rate (C_{cr}), total generations (G), and experiment names.

N	C_w	C_m	C_{cr}	G	Exp. Name
8	50.00%	50.00%	0.00%	124	Exp-01
	50.00%	25.00%	25.00%	124	Exp-02
	75.00%	25.00%	0.00%	247	Exp-03
	75.00%	12.50%	12.50%	247	Exp-04
16	50.00%	50.00%	0.00%	62	Exp-05
	50.00%	37.50%	12.50%	62	Exp-06
	75.00%	12.50%	12.50%	122	Exp-07
	75.00%	18.75%	6.25%	122	Exp-08
32	75.00%	12.50%	12.50%	59	Exp-09
	75.00%	18.75%	6.25%	59	Exp-10
	87.50%	6.25%	6.25%	118	Exp-11
	87.50%	9.37%	3.12%	118	Exp-12
64	87.50%	6.25%	6.25%	55	Exp-13
	87.50%	9.37%	3.12%	55	Exp-14
	93.75%	3.12%	3.12%	110	Exp-15
	93.75%	4.68%	1.56%	110	Exp-16

This process repeats until the final generation Ω^G, where the best individual $\Theta^G = \Theta^{1,G}$ represents the algorithm output (Fig. 2.1):

$$\Omega^G = \{\Theta^{1,G}, \ldots, \Theta^{N,G}\} \text{ with } S^{1,G} \leq \cdots \leq S^{N,G}.$$

2.2.6 Algorithm Parameters: Analysis and Selection

Our objective is to identify parameter configurations that improve both convergence speed and solution quality. This requires careful choice of population size (N), crossover rate (C_{cr}), mutation rate (C_m), and selection rate (C_w).

Larger values of C_{cr} and C_m encourage exploration of the solution space, yielding a more diverse population and increasing the chance of discovering novel solutions. Smaller values emphasize exploitation, focusing on refinement of promising candidates. The selection rate C_w determines how many individuals pass directly to the next generation; higher values favor fitter individuals and can accelerate convergence.

After defining the parameter ranges, the next step is to select the most suitable configuration. Tab. 2.1 summarizes the parameter search, where various combinations are tested to evaluate their impact on performance. In this approach, exploration is limited to a maximum of 500 paths, which guides the parameter selection.

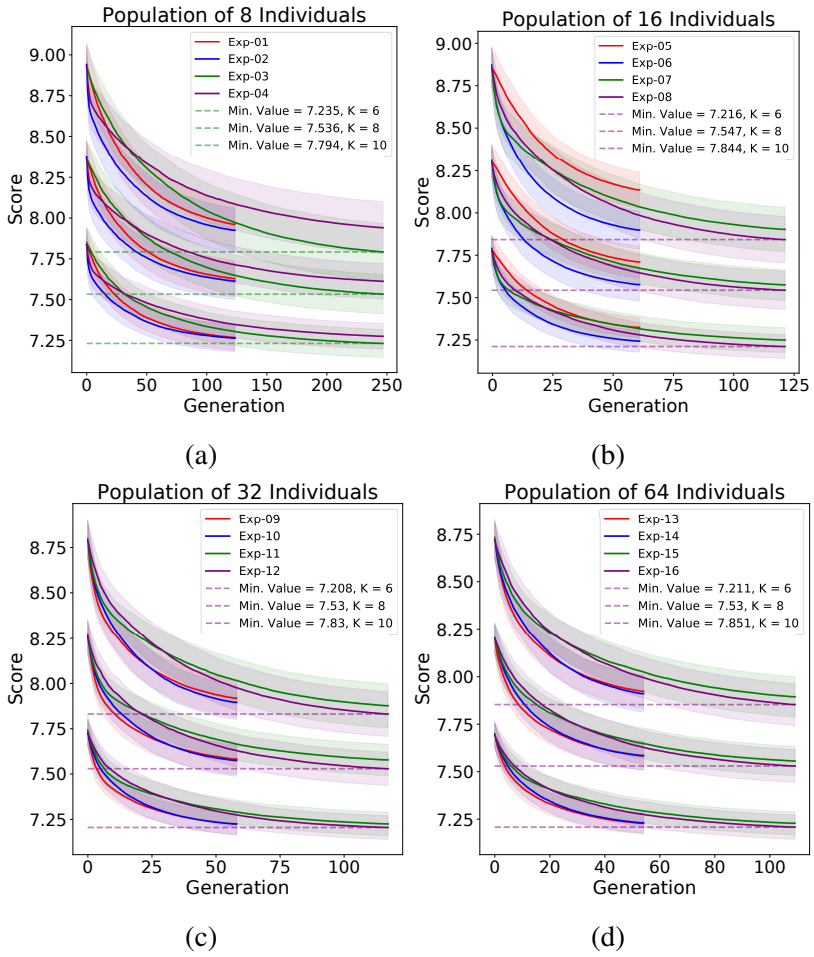


Figure 2.2 Score means and standard deviations for experiments with different population sizes (N) and factors (C_w, C_m, C_{cr}) to reach the minimum fitness value. Each experiment is run with three spring constants: $K = 6$, $K = 8$, $K = 10$ from bottom to top in each plot.

A simulated experiment (Fig. 2.2) is also run with springs fixed at a point $C = [0.6, 0, 0]$, with the path starting at $A = [0, 0, 0]$ and ending at $B = [0.6, 0.6, 0.6]$, using spring constants $K = 6, 8, 10$, maximum force $F_{max} = 30N$, and final exploration ratio $\sigma^G = 0.01$. Each configuration from Exp-01 to Exp-16 is repeated 200 times. These simulations reveal how the algorithm behaves under different settings and support the selection decision.

The plots in Fig. 2.2 compare parameter combinations with respect to minimizing the fitness function. Each experiment uses a different number of individuals (N) and factor values (C_w, C_m, C_{cr}). Overall, Exp-12 with $N = 32$ individuals yields the best convergence and solution quality among the tested options, showing both faster convergence and more consistent reduction of the fitness score. The final parameter choice was based on a careful

Table 2.2 Exp-12 parameter configuration, including number of individuals (N), selection rate (C_w), mutation rate (C_m), crossover rate (C_{cr}), and total generations (G).

N	C_w	C_m	C_{cr}	G	Exp. Name
32	87.50%	9.37%	3.12%	117	Exp-12

trade-off between exploration and exploitation, and this configuration achieves a good balance between searching the fitness landscape and converging toward high-quality solutions.

2.3 Experimental Validation under Elastic Interaction

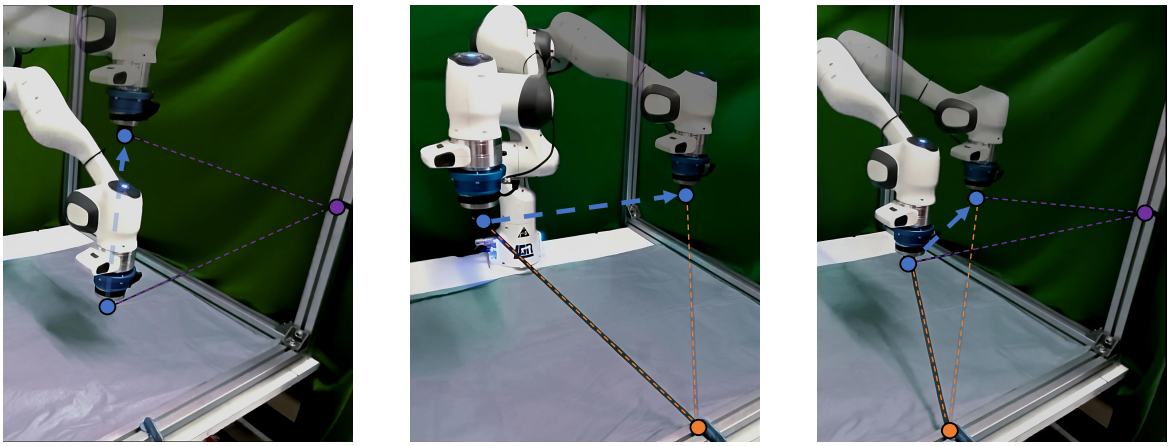
This section outlines the experimental setups and the tests carried out. To evaluate the proposed algorithm, a robotic framework is developed that emulates elastic behavior using a collaborative robot and elastic cords. Specifically, the Exp-12 parameter set shown in Tab. 2.2 is replicated, and the final exploration ratio is set to $\sigma^G = 0.01$.

2.3.1 Experiment Setup

The robot used is a collaborative 7DoF Panda from Franka Emika. An external force/torque sensor from Nordbo is securely attached to the end-effector to measure forces during path execution. The sensor includes a metallic element for holding the elastic cord, while the other end is fixed to a rigid frame. During the experiments, the robot moves between different initial points A and final points B with the path divided into 5 points T. External forces generated by the elastic cords act on the robot throughout the motion. The objective is to validate the proposed GA by finding paths that minimize the fitness function. Three experiments are therefore carried out with distinct setups (Fig. 2.3); Tab. 2.3 reports the fitness values for each case.

Vertical path finding

In this experiment the elastic cord is attached to the robot's metallic element, and its other end is fixed to a vertical frame. The robot moves from $A = [0.63, 0.11, 0.13]$ to $B = [0.56, 0.15, 0.50]$, with a maximum force $F_{max} = 12N$.



(a) **Vertical** (b) **Horizontal** (c) **Two elastic cords**

Figure 2.3 Three experiment configurations: vertical, horizontal, and two elastic cords path finding. Initial and final points and the elastic cords are highlighted.

Table 2.3 Fitness function values and improvements for the first ($S^{1,1}$) and final ($S^{1,G}$) generations across the three experimental setups.

Experiment	$S^{1,1}$	$S^{1,G}$	Improvement (%)
Vertical	9.944	8.399	15.53%
Horizontal	9.606	7.722	19.61%
Two elastic cords	11.153	9.711	12.93%

Horizontal path finding

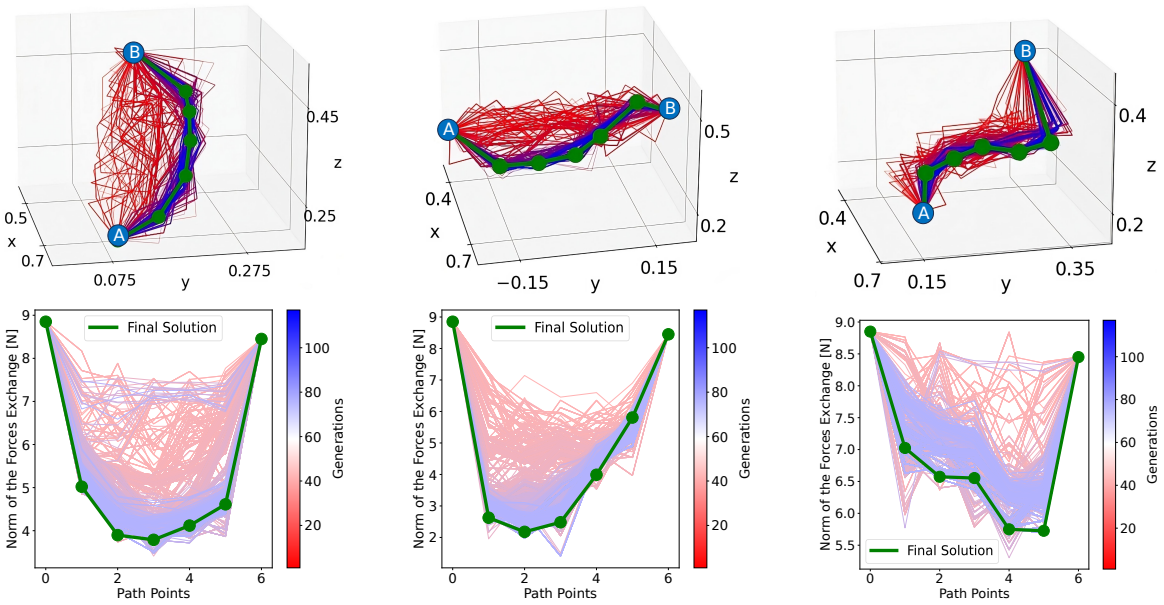
In the second test, the cord end not connected to the robot is fixed to a horizontal frame ($A = [0.5, -0.25, 0.35]$, $B = [0.5, 0.25, 0.35]$, $F_{max} = 12N$).

Two elastic cords path finding

The final experiment combines the previous setups, using two elastic cords: one attached to the vertical frame and the other to the horizontal frame ($A = [0.45, 0.20, 0.13]$, $B = [0.46, 0.34, 0.43]$, $F_{max} = 18N$).

2.3.2 Path Improvement Across Generations

After completing the experiments, we analyze the final paths and the forces exchanged with the elastic cords. In the vertical path finding test (Fig. 2.4a), the path tends to bend toward the attachment point while remaining close to the straight line between the start and end points. The corresponding force profiles indicate a clear reduction along the path from A to B.



(a) Vertical path finding

(b) Horizontal path finding

(c) Two elastic cords path finding

Figure 2.4 Evolution of paths (top) and force exchange with the elastic cords (bottom) across generations for each experiment.

A similar trend appears in the horizontal path finding experiment (Fig. 2.4b). The plots in Fig. 2.4c show the case with two cords, where the path must balance two external forces. Overall, the resulting solutions behave as expected and yield improved fitness values.

Regarding fitness improvement, the smallest gain is observed in the two elastic cords experiment, with a 12.93% improvement, likely due to the difficulty of compensating two forces from different attachment points. In such situations, a compromise path is the natural outcome. The horizontal experiment yields the largest improvement at 19.61%. These results are summarized in Tab. 2.3.

2.4 Discussion and Methodological Implications

In conclusion, a GA-based method is proposed for robotic path optimization in the presence of elastic interaction. The experiments show that, in the tested setups with elastic cords, the approach can reduce the forces exchanged between the robot and the environment while preserving feasible paths. The parameter selection process identifies configurations that improve convergence speed and solution quality in the reported experiments. The method is

model-free and, in this chapter, terminates after a predefined number of generations rather than through a formal convergence guarantee.

The present evidence should nevertheless be interpreted conservatively. The chapter does not establish a general path-planning solution for arbitrary deformable objects or unstructured environments. Rather, it provides controlled proof-of-concept evidence that force-aware evolutionary optimization can be effective when elastic interaction dominates task quality and when sensing is available online. Within the dissertation, this chapter therefore supports the broader methodological claim that sensing-driven adaptation is a practical response to structured interaction uncertainty.

Chapter 3

System-Level Manipulation under Interaction Uncertainty

3.1 Application Context and Automation Challenges

Robotics research increasingly targets the automation of intricate operations, particularly those that still rely on human skill because they demand precision and dexterity. Within manufacturing, wiring harness production is especially relevant: demand is rising and the process requires careful handling of deformable linear objects (DLOs) such as wires and cables. Although the work is repetitive and labor-intensive, it is still largely manual (Nguyen et al. (2020); Olbrich and Lackinger (2022); Trommnau et al. (2019)), making it a compelling candidate for robotic automation.

Advances in sensing now enable systems that can perform these tasks autonomously or in support of human operators, reducing workload. Multi-modal sensor suites that combine tactile, proximity, vision and related modalities provide richer perception, enabling safer interaction with the environment and more accurate execution.

Automation pressure in aerospace and automotive manufacturing, where harness production is critical and time-consuming (Trommnau et al. (2019)), reinforces the importance of this topic. Because DLOs have many degrees of freedom and their deformation is highly nonlinear (Zhu et al. (2022)), recent studies investigate multiple aspects of DLO manipulation. A dynamic model based on discrete elastic rods has been applied to control that arranges DLOs in a desired shape (Lv et al. (2022)). A coupled offline/online, data-driven approach has also been proposed to learn a global deformation model (Yu et al. (2023)). Other methods address shaping by modeling DLOs as kinematic multibody systems using a keypoint encoding network (Huo et al. (2022)). Additional works highlight how advanced sensing improves

manipulation: a capacitive tactile sensor has been used to align fingers with the cable during routing (Monguzzi et al. (2023)), and a tactile gripper has been designed to slide along a cable regardless of material or thickness without losing contact (She et al. (2021)). Nevertheless, integrated solutions that combine these technologies into a fully autonomous wiring harness assembly system remain scarce.

Automation is also limited by the high cost and low adaptability of conventional industrial machines (Hu et al. (2008)) and by the large variety of harness designs. Consequently, manufacturers have explored robotic systems that assist human operators (Heisler et al. (2020); Palomba et al. (2021)), since collaboration is expected to improve productivity (Matheson et al. (2019)). In response, the research community is studying robotic strategies for harness assembly. A robot arm that performs taping while the worker carries out cable positioning and routing has been presented (Román Ibáñez et al. (2021)), whereas a CAD-based solution for aerospace that links wire routing with the CAD design phase has also been proposed (González Huarte et al. (2023)).

This chapter presents a robotized system for automatic wiring harness assembly that integrates state-of-the-art actuation and multimodal sensing. The system consists of two robots: one manages wire positioning and routing using robotic fingers with tactile and proximity sensors to manipulate cables and monitor applied forces, and the other tapes the branches using a purpose-built taping gun. An external depth camera checks task correctness in real time to ensure compliance with specifications. This multimodal sensing, together with robotic manipulators and a highly modular control architecture, improves accuracy and reliability and allows adaptation to different operating conditions and harness variants, which is crucial in this field (Trommnau et al. (2019)). The design also aligns with ongoing efforts to develop robotic cells that can operate alongside humans.

The chapter is organized as follows. Section 3.2 introduces the task and contrasts manual and automated procedures. Section 3.3 describes the hardware and methods. Section 3.4 reports experiments and results. Section 3.5 concludes with limitations and future directions.

3.2 Sensing-Integrated System Architecture

This chapter presents an automated wiring harness assembly system employing two robotic manipulators and custom end effectors, namely fingers with tactile and proximity sensors and a dedicated taping gun. Figure 3.1 shows a 3D representation of the setup together with the reference frames used in the following sections.

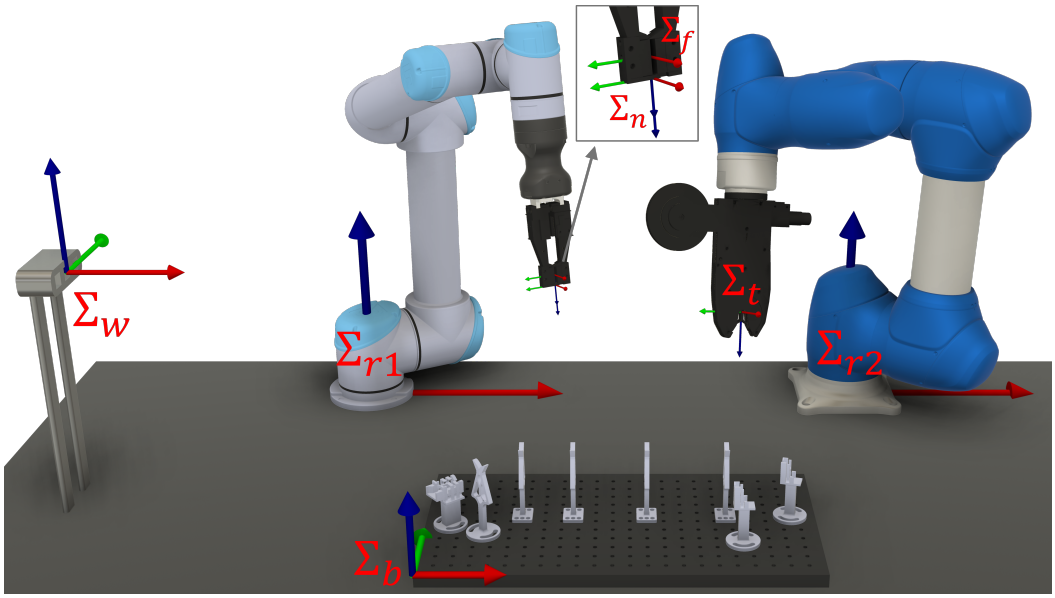


Figure 3.1 3D representation of the robotic system with the reference frames used.

The overall assembly can be viewed as a repetition of three subtasks: branch pick-and-place, cable routing, and spot taping.

3.2.1 Cable Pick-and-Place

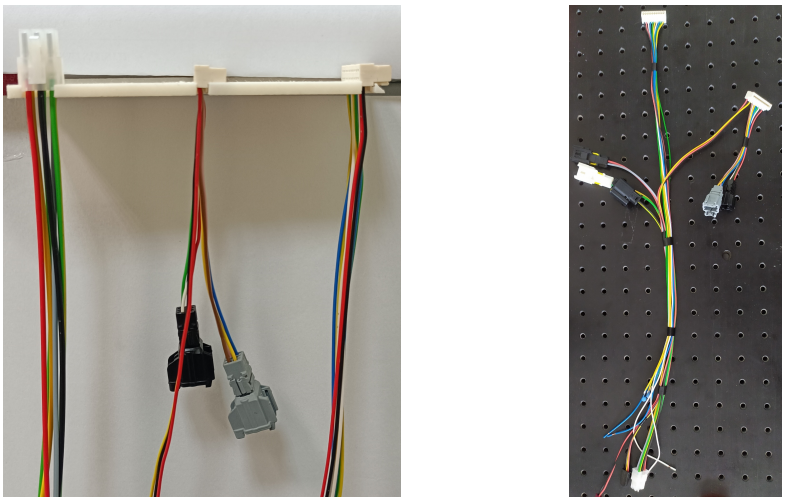
The assembly process starts by retrieving a branch from a warehouse, typically a box or container coming from the previous production step. After retrieval, the branch is placed on the jig board by locking the connector at one end into a specific holder.

For automation, the branches are hung in a dedicated warehouse so that the robot can easily pick them up. A mock-up of this warehouse is shown in Fig. 3.2a. After picking, the robot inserts the connector into a suitably designed holder to secure it in position. During this step, the tactile sensor checks correct insertion of the connector in its holder.

3.2.2 Cable Routing

Once a branch is fixed on the jig board, the individual wires must be routed along the prescribed path. In the manual workflow, a trained worker separates the wires and locks them at specific points using pins, clips, or jigs fixed on the working plane.

In the automated task, this operation is executed by the robotic arm, which computes the required trajectories from the 3D poses of the clips placed on the working plane to respect



(a) (b)

Figure 3.2 Cables stored in the dedicated warehouse (a) and the fully assembled wiring harness (b).

the product specification. The complete trajectory is divided into sub-trajectories computed by different modules of the system, as detailed in Section 3.3.7.

3.2.3 Spot taping

After routing, tape is applied at specific points of the wiring harness, an operation referred to as spot taping. A robotic arm carries a functionally equivalent device, designed to be mounted as a robot end effector and controlled to synchronize taping with the other operations. Details of the custom taping gun are reported in Section 3.3.6.

3.3 Execution Modules and Sensing Integration

Here the methodology and hardware assembled for automatic wiring harness assembly are outlined. The system uses two arms and the strategy is robot-agnostic; the approach is validated on two configurations, one with two UR5e manipulators from Universal Robots and another with a UR5 paired with a Doosan Robotics M0609. The only requirement is the ability to command desired joint positions. A parallel gripper is mounted on one arm. The remaining components and developed methods, beyond off-the-shelf hardware, are described below.

3.3.1 System-Level Contribution

The contribution of this chapter is not limited to assembling hardware modules in a single cell. The key system-level element is an execution loop in which sensorization actively shapes robot behavior under uncertain interaction. Tactile and proximity signals are used online to validate insertion, routing, and manipulation phases; depth-based perception verifies operation correctness; and the planner adapts action sequencing through a modular architecture. In this design, sensing is not a post-hoc monitoring layer but a control-relevant component that changes decisions during execution. This is the primary reason why the system can maintain reliable performance across harness branches with different geometric and interaction constraints.

3.3.2 Sensorized fingers

The fingers adopted in this chapter are a slimmed-down version of those in (Cirillo et al. (2021)), modified to fit the limited space on the assembly board, and they integrate both tactile and proximity sensing. The tactile module uses 12 photo-reflectors arranged in a 6×2 array. Each element consists of a matched LED and phototransistor pair. The array is covered by a silicone pad, and its deformation is detected by the photo-reflectors. The spatial resolution is 3.55 mm, yielding an overall sensitive area of $\approx 22.6 \times 8.4\,\mathrm{mm}^2$.

Proximity sensing relies on the VL6180X Time-of-Flight IR sensor located at the top of the finger. The proximity module sits in a mechanical part with small teeth (fingernails) that separate wires and support them during sliding motions. Figure 3.3 provides an exploded view of the sensorized finger and its components. In this task two fingers are used, but one is passive and contains no sensing elements.

3.3.3 Tactile indicators

From tactile data a contact indicator is computed that relates to the tangential force on the fingers. For the 6×2 taxel array, the indicator is the vector $\mathbf{I}_c = (I_x, I_y)$ with components

$$I_x = x_c - x_{c0}, \qquad I_y = y_c - y_{c0}, \tag{3.1}$$

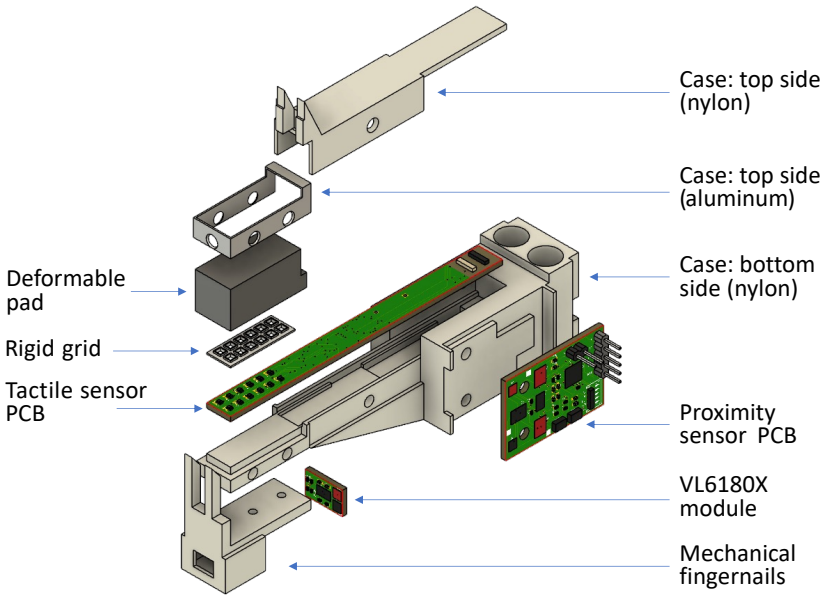


Figure 3.3 CAD view of the finger with the tactile sensor components indicated.

where (x_c, y_c) are the centroid coordinates during contact and (x_{c0}, y_{c0}) correspond to the rest position. The centroid is computed as

$$x_c = \frac{\sum_{i=1}^{12} v_i x_i}{\sum_{i=1}^{12} v_i}, \qquad y_c = \frac{\sum_{i=1}^{12} v_i y_i}{\sum_{i=1}^{12} v_i}, \tag{3.2}$$

where v_i are the voltages and (x_i, y_i) are the mechanical coordinates of the i-th taxel. During operations the norm $\|\mathbf{I}_c\|$ is used as the indicator.

3.3.4 Connector holders

Connectors are secured on the workbench using dedicated holders. These holders were 3D printed with snap-fit tabs that flex during insertion and prevent accidental removal. FEA was employed to estimate insertion forces. For the 6-pole connector, the minimum force to bend each tab by about 3 mm was

$$\|F^{lt}\| = \|F^{rt}\| = 10.6 \text{ N}, \tag{3.3}$$

and, because of the tab inclination, the insertion force along the y axis was estimated as

$$F_{insertion} \simeq 15 \text{ N}. \tag{3.4}$$

Measured slippage on the tactile sensor occurs around $\approx 28\,\mathrm{N}$, so the designed insertion forces remain compatible with safe grasping (Fig. 3.4).

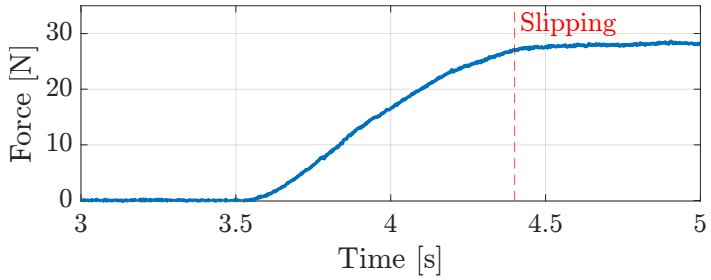


Figure 3.4 Pull test force highlighting slippage at about $\approx 28\,\mathrm{N}$.

3.3.5 Routing clips

The routing clip has a curved profile with a V-shaped cable entry, a retention tab, and deformable compartment tabs. This geometry eases insertion while preventing accidental release; see Fig. 3.5.

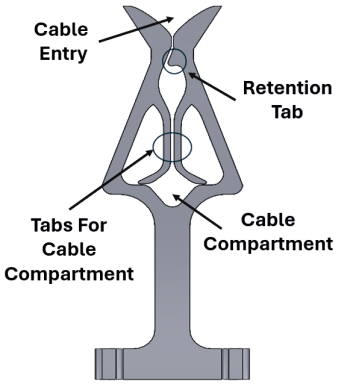


Figure 3.5 Routing clip geometry details.

3.3.6 Taping gun

A compact taping gun suitable for robot mounting was developed. The tool is largely 3D printed, driven by a DC motor with a worm gear, and controlled by an embedded Arduino Nano. The mechanism rotates a C-shaped cable holder twice to apply and cut the tape. The mass was reduced from about 2 kg to below 0.5 kg for robotic integration. Figures 3.6 and 3.7 show the device and its internal mechanism.

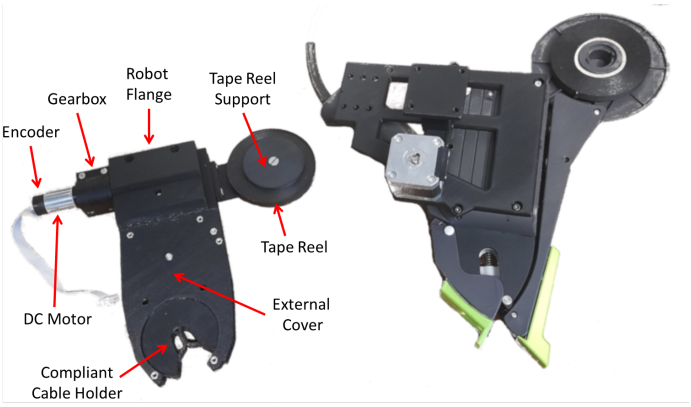
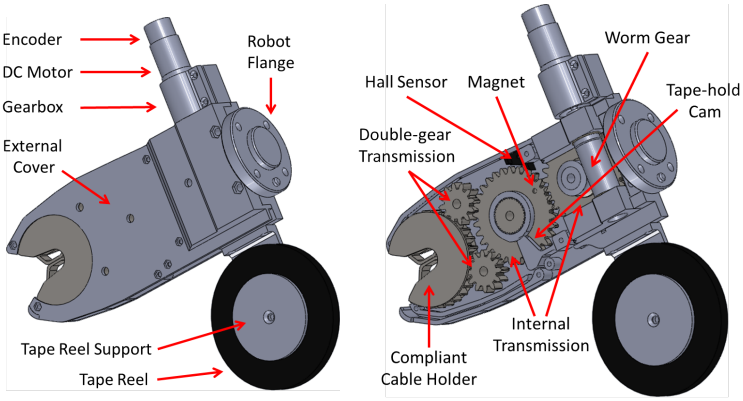


Figure 3.6 Size comparison of the manual taping gun and the robotic version.



(a) (b)

Figure 3.7 Taping gun CAD views: external (a) and internal (b).

3.3.7 Task trajectories

Since wiring harnesses are highly customized, the process is implemented in a modular way. The trajectory generator relies on user-defined configuration files that include CAD information, component information and the sequence of operations. From the CAD input, the planner builds MoveIt planning scenes and orchestrates actions from the sequence file. The overall scheme is shown in Fig. 3.8.

Core actions include pick-and-place, grasping, separation, routing, pushing and taping. For grasping, the z coordinate is computed as

$$P_g = -0.5 \times l + e + p \times n, \tag{3.5}$$

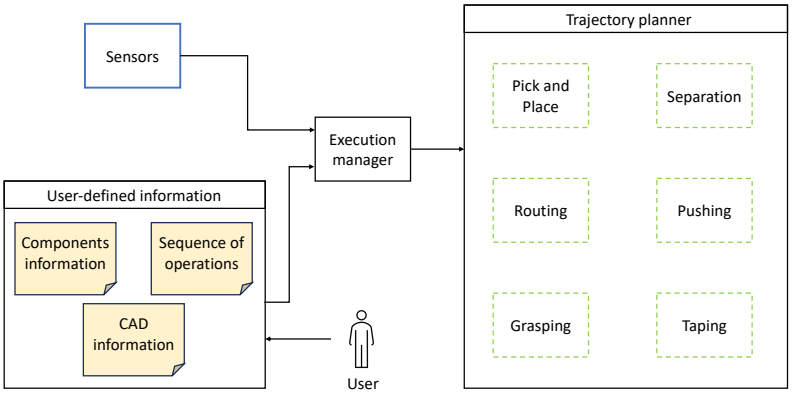


Figure 3.8 Overview of the proposed trajectory generator.

where l is connector length, e the offset to the first wire, p the pitch and n the number of wires to grasp. Routing is performed via linear segments followed by a circular motion to keep wires taut during insertion; see Figs. 3.9 and 3.10.

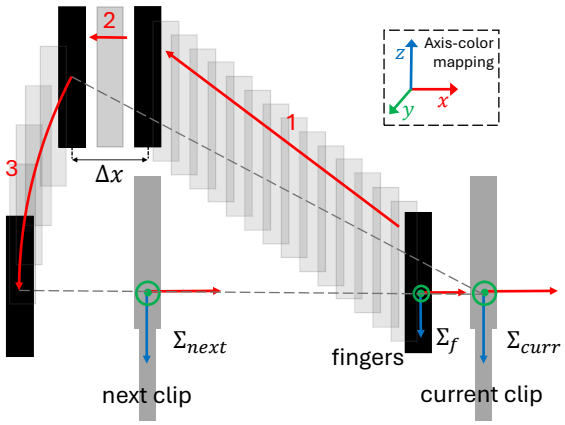


Figure 3.9 Side view of the routing operation: two linear moves followed by a circular motion.

3.3.8 Visual inspection

Visual inspection uses a Zivid One depth camera to acquire point clouds that are classified by a shared-encoder network based on PointNet++ with clip-specific decoders. The encoder is shared and fixed, while a small decoder is trained for each clip to scale the approach. Crops of 10×10 cm around the clips are normalized and rotated to a canonical orientation before inference. The network architecture is shown in Fig. 3.11.

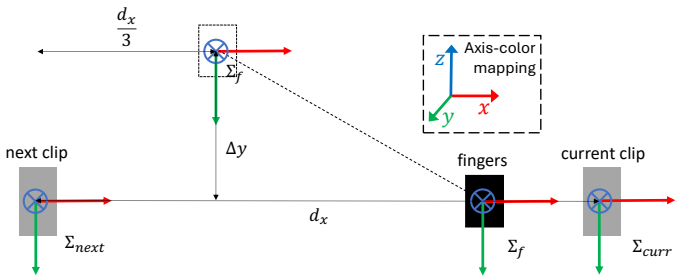


Figure 3.10 Top view illustrating the optional lateral movement during routing.

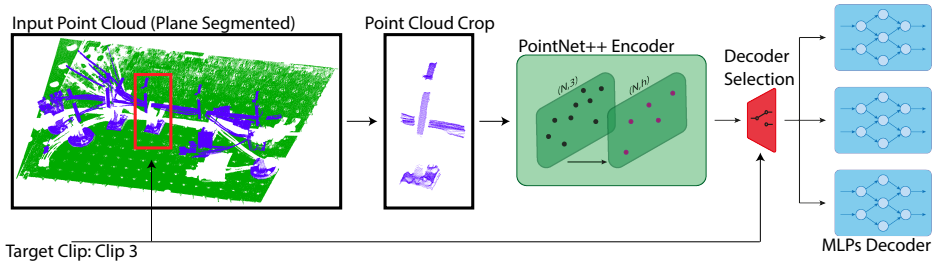


Figure 3.11 Detection pipeline: crop, PointNet++ encoder, then clip-specific decoder.

3.4 Experimental Evaluation and Failure Analysis

Experiments were carried out on an automotive wiring harness comprising three branches with connectors MO6, LU11 and LU10. The evaluation included connector insertion characterization, assembly runs, and visual inspection training and validation.

3.4.1 Connector insertion

FEA predictions were validated using a 6-axis force/torque sensor. For LU11 insertion, the peak force before locking was about 10 N, in line with design expectations. Tactile thresholds were set to 0.2 mm for LU11, 0.25 mm for LU10 and 0.3 mm for MO6. Figures 3.12 and 3.13 show the force and tactile indicator trends.

3.4.2 Assembly task

The planner is implemented as ROS action servers using MoveIt. The sequence file encodes operations as objects of type PC, G, S, R, PD and T. Figure 3.14 shows an example sequence for LU11. The complete assembly sequence was repeated 30 times on two UR5e robots, yielding 24 runs in which all three branches were assembled successfully and 6 runs with at least one branch-level failure, for an overall end-to-end success rate of 80%. The branch-wise

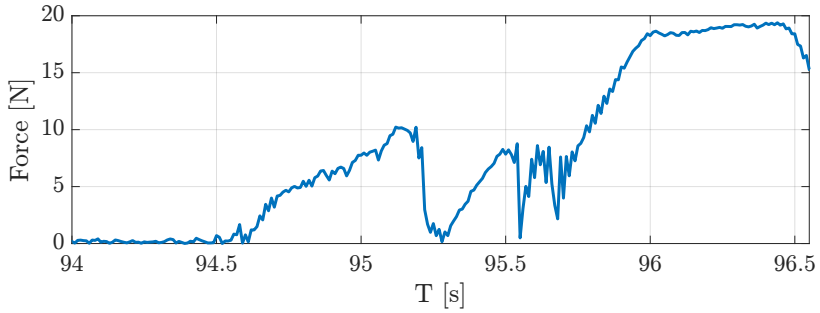


Figure 3.12 Force recorded during LU11 connector insertion.

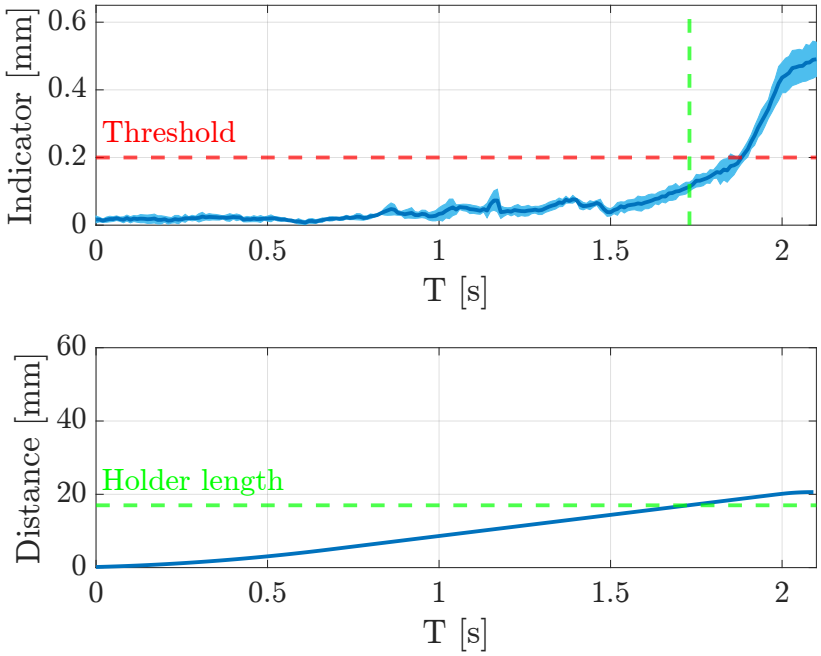


Figure 3.13 LU11 tactile indicator and end-effector travel during insertion.

success rates refer instead to individual branch operations aggregated across all runs: 97% for MO6, 93% for LU11, and 89% for LU10. These branch-level percentages are therefore not directly comparable with the 24/30 end-to-end figure, because the latter counts a run as successful only when the full harness sequence succeeds.

The observed failures were mainly associated with contact-sensitive phases rather than gross motion planning. In particular, the most frequent failure modes were imperfect connector insertion under local pose uncertainty, wire slip during routing transitions near clips, and occasional tolerance accumulation in multi-step branch handling. These cases indicate that the current bottleneck is not trajectory generation per se, but interaction robustness under small geometric and compliance variations. This analysis motivates future extensions toward

adaptive contact thresholds, uncertainty-aware insertion strategies, and tighter coupling between tactile events and replanning policies.

```
- type: PC
  name: Place LU11 connector
  connector: LU11
  vel_max: -0.01
  indicator_threshold: 0.20

- type: G
  name: Grasp 3 LU11 wires
  pick_from: LU11
  wires: 3
  proximity: true

- type: PD
  name: Push down 3 LU11 wires
  clips: [clip5, clip4, clip3, clip2]
  side: [-1, -1, -1, -1]

- type: T
  name: First taping
  clips: [clip2]
  x_offset: [-0.03]
  z_offset: [0]
  rotation: [-10.0]

- type: R
  name: Route 3 LU11 wires
  pick_from: LU11
  up_hole: True
  clips: [clip2,clip3,clip5]
  offset: [0.0,0.08,0.0]
  rotation: [10.0,0.0,0.0]
  lateral: [0,-1,0]
```

Figure 3.14 LU11 cable manipulation operation sequence.

3.4.3 Visual inspection evaluation

Training data are cropped point clouds augmented with Gaussian noise $\mathcal{N}(0, 0.0033)$. The encoder latent dimension is 128, and AdamW was used with a learning rate of $1e{-}4$ and batch size 8. Table 3.1 summarizes precision, recall and accuracy for several training strategies. Pretraining the encoder on related but different data enabled fast decoder fine-tuning with comparable performance.

Table 3.1 Comparison of results using encoders trained on different data sources. Px denotes the number of pre-training epochs. FTx denotes the number of fine-tuning epochs.

Model Type	Training	Precision	Recall	Accuracy
Shared-Encoder	P0 + FT200	0.89	0.95	0.89
	P100 + FT100	0.95	0.97	0.95
	P100 + FT200	**0.95**	**0.99**	**0.96**
Shared-Encoder (Dif.Data)	P100 + FT100	0.92	0.98	0.93
	P100 + FT200	0.94	0.97	0.94

3.5 System-Level Methodological Discussion

This chapter focuses on wiring harness assembly, which is still performed manually even though the operations are highly repetitive. Automation is limited by the complexity of handling electrical wires and by the wide variability of harness designs, making specialized

automatic machines uneconomical. To address this, a robotic system is introduced that can be adapted to different harness types. Adaptability comes from robotic manipulators, custom tools, and easily reconfigurable control software that governs the assembly operations. The resulting architecture integrates dual-arm manipulation, sensorized fingers with tactile and proximity feedback, a robot-mountable taping gun, and depth-camera inspection to support the full workflow from branch handling to routing and spot taping. Experiments on an automotive harness with three branches, different connectors, and varied wire counts provide system-level evidence that the approach is feasible and that the main bottlenecks are concentrated in contact-sensitive phases rather than in gross trajectory generation. Beyond feasibility, the chapter shows that uncertainty-aware execution can be realized as a system property: sensing signals are used to actively shape manipulation decisions, not only to log outcomes. This positions the architecture as an enabling integration contribution to DLO automation within the broader dissertation framework on manipulation under interaction uncertainty. The chapter does not claim a fundamentally new planning algorithm; its main contribution is the sensing-integrated execution architecture and the evidence that such integration materially affects reliability.

Chapter 4

Modular Human-in-the-Loop Programming for Industrial Manipulation

4.1 From System Integration to Human-Guided Programming

This chapter bridges the system-integration perspective of Chapter 3 and the probabilistic shared-autonomy contributions that follow. The focus shifts from building a sensing-rich industrial manipulation cell to structuring how a human can directly program and adapt robotic behavior within that cell through neuromuscular and multimodal interaction. Within the dissertation, the chapter contributes a modular human-in-the-loop design pattern for industrial task programming, together with an experimental validation on robotic connector assembly. Its main limitation is that the evidence is still centered on a representative but bounded industrial case study, rather than on a broad family of collaborative assembly tasks.

Industrial manufacturing is increasingly characterized by high product variability, short life cycles, and frequent reconfiguration of production systems. In this context, collaborative robots (cobots) have emerged as a key technology to support flexible and human-centered automation, enabling humans and robots to safely share workspaces and tasks (Bdiwi et al. (2017); Villani et al. (2018)). Extensive surveys have highlighted the potential of human–robot collaboration (HRC) to improve productivity and ergonomics, while emphasizing the importance of safety and intuitive interaction in industrial settings (Ajoudani et al. (2018)).

More recent studies further stress the need for proactive and adaptive collaboration strategies to cope with dynamic industrial environments (Johannsmeier and Haddadin (2019, 2021)).

Despite these advances, the deployment of collaborative robots in real industrial scenarios is still hindered by high integration and programming costs. In particular, frequent changes in products, layouts, and processes require repeated reprogramming and validation, often relying on expert knowledge (Villani et al. (2018)). These challenges are exacerbated by stringent safety requirements associated with physical human–robot interaction. Fundamental work on collision detection and reaction has laid the foundations for safe interaction (Haddadin et al. (2008)), while subsequent studies have addressed safety evaluation through crash testing (Haddadin (2015)) and comprehensive analyses of physical safety in robotics (Haddadin (2016)). Industrial deployment is further guided by international safety standards such as ISO/TS 15066 (International Organization for Standardization (2016)).

Beyond physical safety, effective human–robot collaboration is inherently a socio-technical problem. It requires predictability, transparency, and mutual understanding between human operators and robotic systems (Ajoudani et al. (2018)). Fully autonomous solutions, while appealing in principle, often struggle to cope with task variability and unforeseen situations typical of industrial environments. Conversely, pure teleoperation and low-level programming impose a high cognitive burden on operators and limit scalability.

To address this trade-off, Human-in-the-Loop (HiTL) paradigms have gained increasing attention. In HiTL systems, control authority is dynamically shared between human and robot, allowing humans to provide high-level intent and corrections while robots handle low-level execution. This concept has been formalized in shared control frameworks through policy blending (Dragan and Srinivasa (2013b)), and extended by learning-based approaches that leverage deep reinforcement learning to improve shared-autonomy behavior under uncertainty (Reddy et al. (2018)). In industrial assembly, intention tracking methods have been proposed to infer human goals and support robust collaboration (Huang et al. (2023)). The integration of multimodal sensory data has further been shown to enhance human-centered robot control in collaborative manufacturing tasks (Johannsmeier and Haddadin (2020)), including connector assembly scenarios based on vision and tactile sensing for DLO manipulation and pin insertion (Caporali et al. (2026)).

The effectiveness of Human-in-the-Loop collaboration strongly depends on the quality of the interaction channels through which humans and robots communicate. Traditional industrial interfaces, such as teach pendants, are poorly suited for intuitive interaction. As a result, multimodal human–robot interfaces have been widely investigated. Gesture- and speech-based approaches have demonstrated effective robot teleoperation and task

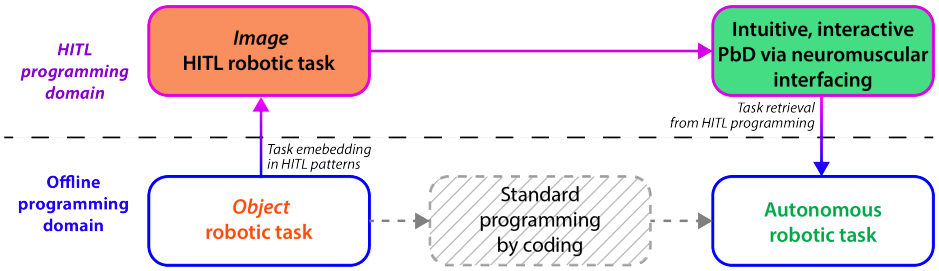


Figure 4.1 Conceptual workflow of the proposed HITL programming paradigm. The object robotic task is mapped from the offline programming domain to an HITL programming domain based on neuromuscular human–robot interaction, and then translated back to enable autonomous robot execution.

specification (Stiefelhagen et al. (2019)). Hybrid paradigms combining conversational interfaces with block-based programming have been proposed to lower the barrier for non-expert users (Canal et al. (2022)). Other works have explored multimodal programming frameworks that integrate multiple input modalities within a unified interaction pipeline (Pfeiffer and Bdiwi (2020)).

Wearable sensing has further enriched multimodal interaction. In particular, surface electromyography (sEMG) has been exploited to augment kinesthetic teaching and estimate human intent or impedance (Rozo et al. (2016)). Augmented reality approaches have enabled intuitive robot programming, visualization, and validation directly in the workspace (Makris et al. (2017)).

Despite the effectiveness of multimodal Human-in-the-Loop interfaces, most existing solutions are developed for specific tasks and embedded in monolithic system architectures. Such designs limit scalability, reuse, and industrial certification. Recent research has highlighted the benefits of modular robot control architectures based on reusable skills (Pecora et al. (2018)) and structured task representations. Behavior Trees, in particular, provide a modular and reactive formalism for task-level control (Colledanchise and Ogren (2021)), and have been successfully applied to collaborative robot programming (Jain et al. (2024)). These approaches are increasingly adopted within industrial ecosystems such as ROS-Industrial (ROS-Industrial Consortium (2020)).

However, a unifying design pattern that systematically integrates Human-in-the-Loop principles, multimodal interaction, and modular control architectures for industrial collaborative robotics is still missing.

4.2 Human-in-the-Loop Programming Architecture

We consider an industrial robotic system properly designed and integrated to execute manipulation tasks in collaborative manufacturing environments. The focus of this chapter is not on low-level control or hardware design, but on the definition of a high-level HITL programming methodology for intuitive robot manipulation. Rather than relying on traditional programming approaches, such as textual coding or teach pendant interfaces, we aim to enable task programming through direct human interaction, where the user interacts with the robot via a neuromuscular interface based on surface sEMG signals. This objective poses several challenges. sEMG signals are inherently time-varying and affected by non-idealities that are difficult to model explicitly, requiring adaptive decoding strategies. Moreover, effective HITL programming demands the provision of task-relevant feedback to the user beyond visual robot observation, as well as the ability to selectively reprogram specific portions of the task. To address these requirements, a modular HITL design pattern is introduced based on the composition of *HITL patterns*. This paradigm is constituted by specific bidirectional human-robot interaction modules that enable programming capabilities, and can be flexibly adapted to specific scenarios. These modules are coordinated by a structured supervisory layer based on a finite-state machine (FSM) driven representation. The following sections detail the proposed design pattern and its constituent components.

4.2.1 Task Programming through Human-in-the-Loop Interaction

The objective of robot programming in industrial manipulation is to configure a robotic system so that it can autonomously execute a desired task in a reliable and repeatable manner. In general, the task specification can be expressed as an *object robotic task*, i.e., a description of the manipulation goals in terms of objects, actions, and task constraints defined within the operational environment. The programming problem therefore consists in configuring the robotic system so that the desired object-level task can be executed autonomously. In conventional industrial practice, this problem is addressed within what we refer to as the *offline programming domain*. In this paradigm, the user programs the robot through textual coding, scripting languages, or proprietary programming interfaces. The interaction between the user and the robotic system is typically indirect and discontinuous, since programming and execution occur in separate phases. Moreover, the programming process is not inherently intuitive: the user must possess specialized knowledge of robot kinematics, control architectures, and software frameworks. As a result, programming complex manipulation tasks

often requires interdisciplinary engineering expertise and involves iterative trial-and-error procedures before a satisfactory task implementation is achieved.

The approach adopted in this chapter introduces an alternative programming paradigm based on an HITL interaction framework. Rather than directly solving the programming problem within the offline programming domain, the proposed methodology first maps the object robotic task into a different functional domain, where the task is reformulated as an *HITL robotic task*. Within this domain, the programming process takes place through continuous bidirectional interaction between the human operator and the robotic system. More specifically, the user communicates task-related commands to the robot through an intuitive neuromuscular interface based on the decoding of sEMG. At the same time, the robotic system provides feedback to the user regarding the current state of the programming process, the execution phase of the task, and relevant sensory measurements acquired from the robot and the environment. This bidirectional interaction enables the operator to program the fundamental features of the robotic task through intuitive and interactive control actions, without requiring explicit low-level programming. Once the HITL programming phase is completed, the resulting HITL robotic task can be translated back into the original object-level domain. At this stage, the robotic system has acquired all the information necessary to autonomously execute the desired manipulation task. The HITL interaction phase therefore acts as an intermediate programming layer that allows the user to intuitively specify task behaviors while maintaining compatibility with conventional robotic execution frameworks.

The conceptual workflow of this paradigm is illustrated in Fig. 4.1. Starting from the specification of an object robotic task, the proposed approach maps the programming problem into the HITL programming domain, where the task is programmed through bidirectional neuromuscular human-robot interfacing. Once the programming phase is completed, the resulting task representation is mapped back to the offline programming domain, allowing the robot to execute the task autonomously.

In the following sections, we first describe the characteristics of the robotic system and the human–robot interface considered here. Subsequently, we introduce the modular HITL components and the HITL design pattern structure that enable the practical realization of the proposed programming paradigm.

4.2.2 Neuromuscular Interface and Experimental System

In order to instantiate the proposed HITL design pattern, we consider a robotic manipulation system equipped with multimodal sensing and a neuromuscular human–robot interface.

While the design pattern developed in this chapter is general and independent of a specific hardware configuration, here a representative industrial setup is assumed that enables bidirectional interaction between the human operator and the robotic system.

The experimental setup considered here is illustrated in Fig. 4.2. The robotic platform consists of a collaborative manipulator equipped with a sensorized end-effector designed to support contact-rich manipulation tasks. In particular, the end-effector integrates both visual and tactile sensing capabilities, which are used to monitor the state of the task and to enable closed-loop interaction with the environment. A stereo vision module is mounted near the end-effector to estimate the three-dimensional pose of manipulated objects. The stereo system is composed of two high-resolution cameras configured in a converging geometry to improve depth accuracy while maintaining a compact mechanical integration with the gripper. The cameras provide synchronized image streams that are processed through stereo reconstruction algorithms to estimate the three-dimensional structure of the scene and recover the full 6D pose of relevant task features. This visual information is used to compensate for grasp-induced misalignments and to support precise alignment during manipulation and insertion operations. In addition to vision sensing, the gripper fingers are equipped with tactile sensors based on optoelectronic taxel arrays. Each finger integrates a matrix of sensing elements covered by a compliant elastomeric layer. Local deformations of the sensing surface caused by contact with objects or the environment produce variations in the reflected optical signal measured by the taxels. The resulting tactile measurements provide a spatially distributed representation of contact interactions across the finger surface. This information is used to detect contact transitions, monitor interaction forces, and recognize characteristic contact patterns associated with successful manipulation outcomes.

The human operator interacts with the robotic system through a neuromuscular interface. The interface consists of two wearable sEMG acquisition bracelets placed on the two operator's forearms. The first EMG bracelet is used to decode the user's control intentions, which are translated into robot commands through a machine-learning-based intent decoding module. Both discrete and continuous control signals can be generated depending on the programming phase, enabling the specification of task sequences as well as the modulation of robot motion during execution. The second EMG bracelet is dedicated to detecting specific co-contraction patterns of the forearm muscles. These patterns are used as supervisory triggers to switch between different interaction modes, for instance to enter training phases or to initiate reprogramming procedures during HITL operation, as detailed in the following subsections. The operator also wears a vibrotactile feedback device placed on the forearm. This device conveys haptic information regarding the state of the robotic system and the

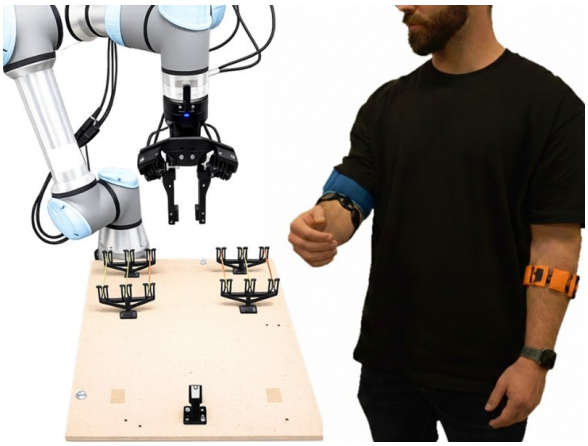


Figure 4.2 Experimental setup used to instantiate the proposed HITL programming paradigm. A collaborative robot equipped with stereo vision and tactile sensing operates on a cable wiring workbench, while the operator interacts with the system through two sEMG acquisition bracelets and a vibrotactile feedback device.

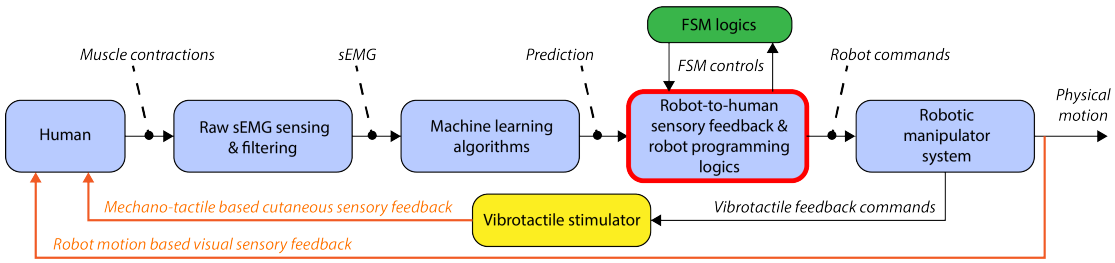


Figure 4.3 Base HITL control architecture underlying the proposed design pattern. The scheme represents the neuromuscular interaction loop between the human operator and the robotic system, including sEMG sensing and filtering, machine-learning-based decoding, robot programming and sensory feedback logics, and vibrotactile feedback generation.

interaction forces measured at the gripper. The vibration intensity is modulated according to sensory signals acquired during the task, allowing the operator to perceive relevant contact events and to regulate the interaction accordingly.

4.2.3 Core Human–Robot Interaction Architecture

The proposed design pattern is grounded on a fundamental HITL control structure that defines the basic interaction loop between the human operator and the robotic system. This base architecture is illustrated in Fig. 4.3. Within the proposed methodology, the object robotic task introduced in the previous subsection is first mapped into this HITL control domain, where robot behaviors can be programmed through intuitive and interactive human–robot interaction. The resulting HITL programming representation is subsequently integrated

within a higher-level supervisory logic based on a finite-state machine (FSM), which will be introduced in the following later in this section.

The HITL base architecture defines a closed interaction loop between the human operator and the robot, where neuromuscular signals are decoded into control commands, and sensory feedback is conveyed back to the operator. The architecture is composed of the following main modules.

Raw sEMG sensing and filtering

The human interface relies on two wearable sEMG armbands placed on the operator's forearms. The raw sEMG signals are processed through a preprocessing pipeline to extract robust muscle activation features. The preprocessing stage includes a 50 Hz notch filter to suppress power-line interference, a 20 Hz high-pass filter to remove motion artifacts, root-mean-square (RMS) feature extraction over a sliding window of 200 ms. Let $e_s(t)$ denote the filtered signal from electrode s. The RMS feature is computed, and the resulting feature vector $\mathbf{e}(t) = [e_1^{\mathrm{RMS}}(t), \ldots, e_8^{\mathrm{RMS}}(t)]^T$ represents the current muscle activation pattern and constitutes the input to the machine learning modules.

Machine learning algorithms

The decoded neuromuscular signals are interpreted through two machine learning modules that operate on the two EMG bracelets.

Intent decoding via incremental regression The bracelet worn on the dominant arm is used to decode the user's control intent. This decoding is implemented through a nonlinear incremental learning model based on incremental ridge regression (RR) (Braun et al. (2025)). Given an input feature vector $\mathbf{x}(t)$ derived from the sEMG signals, the regression model estimates the control output

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

where $\mathbf{w}$ is the learned parameter vector. The optimal parameters are obtained by minimizing

$$\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{2} \sum_{i=1}^{m} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

where λ is a regularization parameter. To capture nonlinear relationships between muscle activation patterns and control commands, the input features are projected into a nonlinear space using Random Fourier Features (RFF). Given a projection vector ω sampled from a

Gaussian distribution and a random phase b, the nonlinear mapping is defined as

$$z_\omega(\mathbf{x}) = \sqrt{2}\cos(\omega^T\mathbf{x}+b)$$

and multiple such features are concatenated to obtain a nonlinear feature vector

$$\mathbf{z}(\mathbf{x}) = \sqrt{\frac{1}{D}}[z_1(\mathbf{x}),\ldots,z_D(\mathbf{x})]^T.$$

This representation allows the incremental regression model to approximate kernel ridge regression while maintaining constant computational complexity.

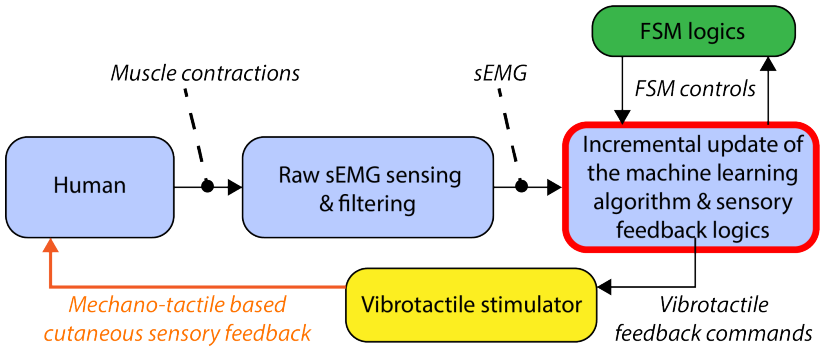


Figure 4.4 HITL incremental (re)training architecture. The FSM can switch the system to a training state, where the regression model used for intent decoding is incrementally updated using new sEMG samples. The incremental update module simultaneously manages model adaptation and vibrotactile feedback to guide the operator during gesture acquisition.

Cocontraction estimation The bracelet worn on the opposite arm is used to estimate the level of muscular cocontraction, which is exploited as a supervisory control signal. Let $\alpha_E(t)$ and $\alpha_F(t)$ denote the dominant neural drives associated with extensor and flexor muscles, respectively. The cocontraction level is defined as

$$\gamma(t) = \min(\alpha_E(t), \alpha_F(t)).$$

The neural drives are estimated through an unsupervised decomposition of the RMS EMG signals using nonnegative matrix factorization (NMF). Let

$$\mathbf{E}(t) = [e_1^{\mathrm{RMS}}(t),\ldots,e_8^{\mathrm{RMS}}(t)]^T$$

be the RMS signal vector. The decomposition model is

$$\mathbf{E}(t) = \mathbf{W} \begin{bmatrix} \alpha_E(t) \\ \alpha_F(t) \end{bmatrix},$$

where $\mathbf{W}$ is a drive-to-EMG mixture matrix estimated offline. The neural drives are then obtained online as

$$\mathbf{A}(t) = \begin{bmatrix} \alpha_E(t) \\ \alpha_F(t) \end{bmatrix} = \mathbf{W}^{+}\mathbf{E}(t)$$

where $\mathbf{W}^{+}$ denotes the pseudoinverse of $\mathbf{W}$.

Human–robot sensory feedback and programming logic

The predictions generated by the machine learning modules are processed by the human–robot interaction logic layer. This module interprets the decoded intents and translates them into commands for the robotic manipulator. Two types of actions can be generated. First, robot motion commands may be issued to produce visible movements of the manipulator, allowing the operator to observe the effect of the programming actions through visual feedback. Second, haptic feedback commands may be generated to activate the vibrotactile interface. In this case, the module can either signal discrete programming events or map physical quantities measured by robot sensors (such as tactile interaction forces) into vibrotactile stimuli perceived by the operator. This block also manages the interaction between the HITL control loop and the supervisory FSM layer, which orchestrates the different programming phases, as explained in the following section.

Vibrotactile stimulator

The vibrotactile stimulator provides cutaneous sensory feedback to the operator. Let $\phi(t)$ denote the vibrotactile command delivered to the operator. Two operating modes are supported. In the first mode, discrete vibrotactile pulses with fixed intensity and configurable duration are used to notify the operator about specific programming events (e.g., command confirmation or phase transitions), resulting in piecewise-constant profiles of $\phi(t)$. In the second mode, the vibration intensity is modulated continuously to encode physical quantities measured by the robotic system. For instance, the vibration amplitude may be mapped proportionally to the level of tactile interaction detected at the gripper. This continuous mapping provides an intuitive perception of contact dynamics during task programming and execution, allowing the operator to regulate the interaction behavior of the robot.

4.2.4 Incremental Retraining Strategy

The HITL interaction loop introduced in the previous subsection enables intuitive robot programming through neuromuscular signals. However, due to the intrinsic non-stationarity of sEMG signals, the intent decoding model must be periodically updated to maintain reliable performance. Variations in muscle activation patterns may arise from electrode displacement, muscular fatigue, or physiological variability across users and sessions. For this reason, the proposed architecture supports online incremental training and retraining of the intent decoding model (Braun et al. (2025)). The retraining architecture is illustrated in Fig. 4.4. The process is activated when the supervisory finite-state machine (FSM) switches the system to a dedicated training state. In this condition, the operator is guided through a sequence of gesture demonstrations that are used to incrementally update the regression model described in the previous subsection. The training interaction is coordinated by the block named *incremental update of the machine learning algorithm & sensory feedback*. This block performs two simultaneous functions. First, it updates the regression model parameters using the newly acquired sEMG samples. Second, it generates vibrotactile feedback signals that guide the operator during the training procedure. When the system enters a training state, the operator receives two short vibrotactile pulses followed by one long pulse, which confirm the successful state transition. After an 8 s preparation interval, the operator receives three short vibrotactile pulses that indicate the start of a new training trial. Immediately afterward, a long vibration of 10 s duration and intermediate intensity is delivered for the entire duration of the gesture acquisition window. During this interval, the operator performs the required muscular contraction pattern, while the corresponding sEMG signals are collected and used to incrementally update the decoding model.

Let $\mathbf{x}(t)$ denote the feature vector extracted from the filtered sEMG signals, as introduced in the previous subsection. The nonlinear feature mapping based on Random Fourier Features is defined as

$$\mathbf{z}(\mathbf{x}) = \sqrt{\frac{2}{D}}\cos(\Omega\mathbf{x}+\mathbf{b})$$

where $\Omega \in \mathbb{R}^{D\times n}$ contains random projection vectors sampled from a Gaussian distribution $\mathcal{N}(0,2\eta)$ and $\mathbf{b} \in \mathbb{R}^{D}$ is a vector of random phases uniformly distributed in $[0,2\pi]$, where η denotes the bandwidth parameter of the Random Fourier Features mapping. The regression model is then defined as

$$\sigma(t) = \mathbf{w}^{T}\mathbf{z}(\mathbf{x}(t))$$

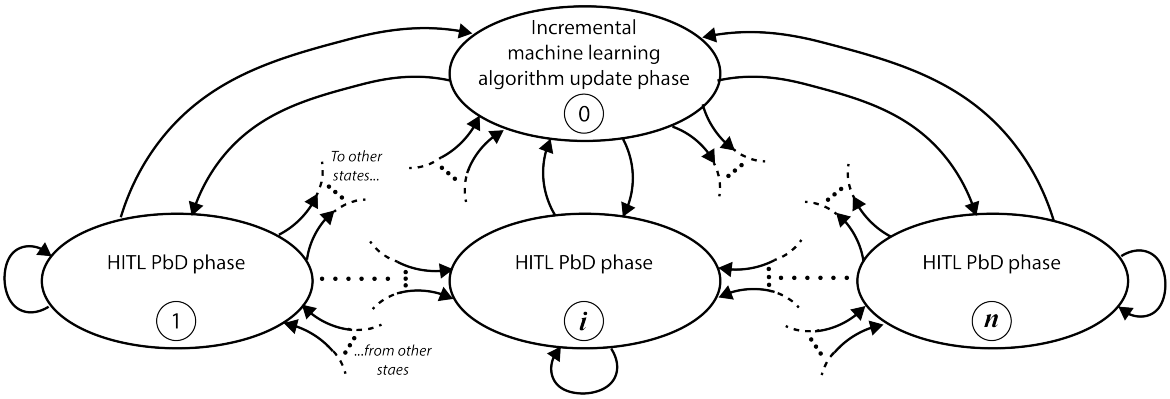


Figure 4.5 Finite-State Machine (FSM) supervising the HITL programming process. Each state activates the HITL interaction framework to program a specific component of the robotic task, while a dedicated retraining state enables incremental adaptation of the EMG decoding model.

where $\mathbf{w}$ denotes the weight vector of the regression model. Model parameters are updated incrementally using the incremental ridge regression formulation. Let

$$\mathbf{A} = \lambda \mathbf{I} + \mathbf{Z}^T \mathbf{Z}$$

be the regularized covariance matrix associated with the transformed features. Instead of recomputing the full inverse matrix, the update is performed using the Sherman–Morrison rank-one update rule. Given a new training pair $(\mathbf{x}, y)$, the update procedure is

$$\mathbf{z} = \mathbf{z}(\mathbf{x})$$

$$\beta \leftarrow \beta + \mathbf{z}y$$

$$\mathbf{A}^{-1} \leftarrow \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{z}\mathbf{z}^T\mathbf{A}^{-1}}{1 + \mathbf{z}^T\mathbf{A}^{-1}\mathbf{z}}$$

$$\mathbf{w} \leftarrow \mathbf{A}^{-1}\beta$$

where λ is the regularization parameter. This formulation allows the regression model to be updated online without storing previous training samples, thus enabling efficient real-time update. During the retraining phase, the incremental update module communicates with the FSM supervisory layer, which orchestrates the sequence of training gestures and interaction phases, as explained in the following subsection.

4.2.5 Supervisory State-Machine Coordination

While the HITL base architecture described in the previous subsections defines the fundamental human–robot interaction loop, complex robot programming tasks typically require the coordination of multiple programming phases. For this reason, the proposed design pattern introduces a supervisory control layer based on a FSM, which structures the overall HITL programming process. The FSM architecture adopted here is illustrated in Fig. 4.5. The key idea is that the overall HITL programming procedure can be decomposed into a sequence of programming states, each associated with the specification of a particular component of the robotic task. Within each state, the HITL base framework introduced previously is activated and configured to program a specific element of the manipulation behavior. Examples of such elements include grasp configuration, motion generation, interaction parameters, or other task-dependent control primitives. In each state, the predictions produced by the sEMG decoding module are mapped into appropriate robot commands, while the operator receives feedback through vibrotactile stimulation and visual observation of the robot motion. More formally, let

$$\mathscr{S} = \{s_1, s_2, \ldots, s_N\}$$

denote the set of FSM states. Each state s_i corresponds to a specific programming phase, in which the HITL interaction loop is configured to control a particular aspect of the robotic task. During state s_i, the decoded EMG prediction $\sigma(t)$ is interpreted according to a state-dependent control policy

$$u(t) = \pi_i(\sigma(t)),$$

where $u(t)$ denotes the robot control command and $\pi_i(\cdot)$ is the mapping associated with state s_i. Similarly, the sensory feedback delivered to the operator is also configured according to the current state, allowing vibrotactile and visual cues to convey information relevant to the specific programming phase. The FSM also includes a dedicated retraining state, which enables the incremental update of the sEMG decoding model described in the previous subsection. Transition to the retraining state is triggered when a sustained muscular cocontraction is detected. Specifically, let $\gamma(t)$ denote the cocontraction level estimated from the sEMG signals. If the condition

$$\gamma(t) > 50$$

is maintained continuously for at least three seconds, the FSM switches to the retraining state, activating the incremental learning procedure illustrated in Fig. 4.4. The cocontraction signal is normalized in the interval $[0, 100]$, where 100 represents the maximum simultaneous

activation of the antagonist muscle groups. Transitions among the standard programming states occur instead when the operator completes the programming of the corresponding task component. For instance, once a grasp configuration has been defined, the FSM advances to the next state responsible for programming the subsequent motion phase. This sequential organization enables the operator to progressively construct the overall robotic task through a series of intuitive HITL programming stages. The FSM therefore acts as a supervisory coordination layer that orchestrates the different HITL interaction modules, ensuring a structured programming workflow while preserving the intuitive interaction paradigm. By separating the interaction logic from the task sequencing, the proposed architecture enables modular task specification, facilitates system reconfiguration, and supports scalable development of complex collaborative robotic applications.

4.3 Connector Assembly Case Study

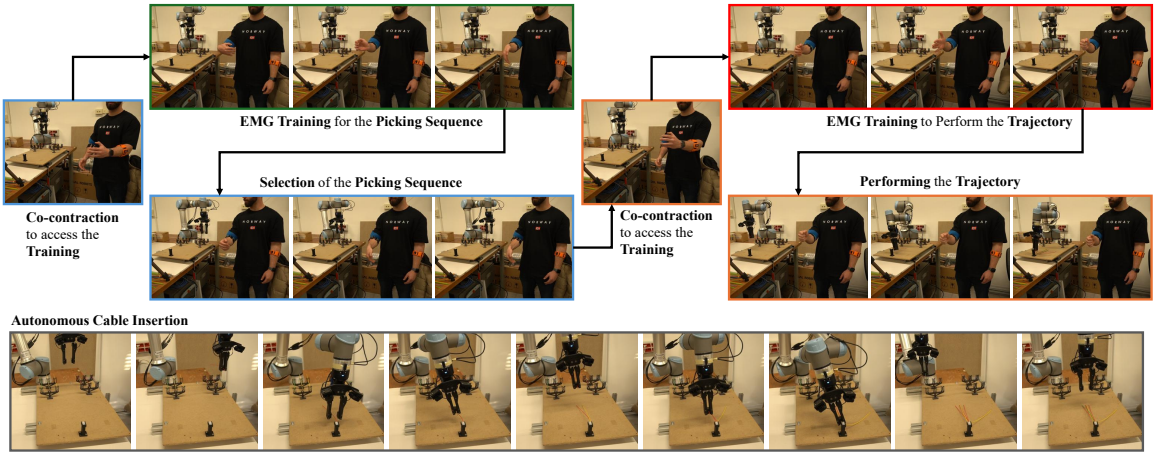


Figure 4.6 General experimental workflow for the robotic connector assembly case study. The upper-left branch reports State 2 (*Picking Sequence Training*) followed by State 1 (*Picking Sequence*); the upper-right branch reports State 4 (*Performing Trajectory Training*) followed by State 3 (*Performing Trajectory*); the bottom row shows the resulting autonomous cable insertion obtained after the two programming stages have been specified.

4.3.1 Case-Study Task

Figure 4.6 summarizes the general scheme of the experiment. To demonstrate the proposed modular HITL design pattern, we consider a robotic connector assembly task performed on the cable wiring workbench described in Section 4.2. The objective is to program a

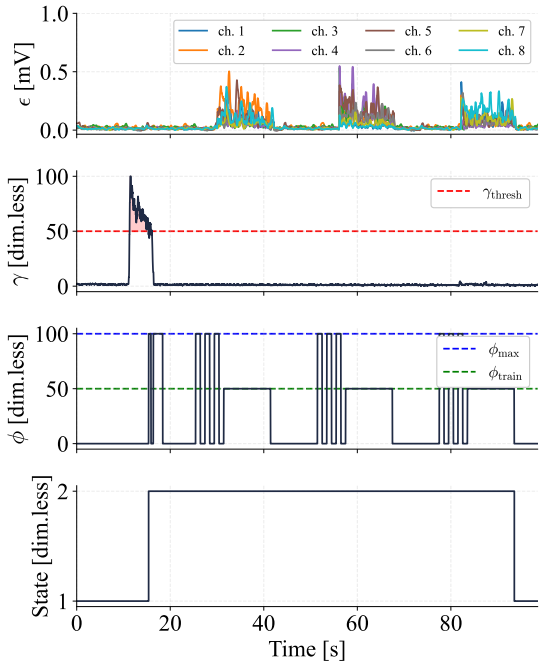


Figure 4.7 State 2: *Picking Sequence Training*. From top to bottom: multichannel sEMG activity e, cocontraction level γ, vibrotactile command ϕ, and FSM state variable. The threshold crossing of γ activates the transition from State 1 to State 2; the vibrotactile profile then provides the state-change confirmation, the warning pulses preceding acquisition, and the 10 s intermediate-intensity stimulation used to maintain each discrete gesture before returning to State 1.

collaborative manipulator to autonomously pick a connector from a set of candidate locations, execute the corresponding approach and transfer motion, and finally perform the cable insertion operation. This application is representative of industrial manipulation scenarios in which the overall task can be decomposed into successive programming phases, each requiring a different interaction policy between the operator and the robot.

The experimental workflow is organized into two main HITL programming stages. In the first stage, denoted as *Picking Sequence*, the operator specifies the discrete sequence of choices needed to select the connector location. In the second stage, denoted as *Performing Trajectory*, the operator programs the continuous robot motion used to carry the selected connector toward the insertion area. Once these two stages have been completed, the robot autonomously executes the resulting connector insertion sequence, as shown in the bottom row of Fig. 4.6.

4.3.2 Case-Study Instantiation of the Design Pattern

The case study instantiates the general HITL architecture described in Section 4.2 through a four-state FSM. State 1 corresponds to *Picking Sequence*, State 2 to *Picking Sequence Training*, State 3 to *Performing Trajectory*, and State 4 to *Performing Trajectory Training*. States 1 and 3 are the operative programming states, whereas States 2 and 4 are the training states used to incrementally adapt the intent decoder before returning to the corresponding online phase.

In State 1, the *Picking Sequence* phase is realized as a discrete interaction mode. In this condition, the intent decoder maps three wrist gestures into symbolic commands used to navigate the sequence definition: wrist flexion is associated with the command *left*, wrist extension with *right*, and wrist adduction with *confirm*. These decoded commands are used to move through the set of admissible picking options and to validate the currently selected one. State 2 is the associated training phase in which the same three gestures are recorded under vibrotactile guidance to update the discrete decoder.

State 3 implements the *Performing Trajectory* phase as a continuous interaction mode. In this phase, the decoded neuromuscular output is used to modulate the robot motion along the task trajectory. The associated State 4 training procedure is performed by recording three reference gesture conditions, namely *rest*, *open*, and *close*. The *open* gesture is associated with backward robot motion, whereas the *close* gesture is associated with forward robot motion. The *rest* condition corresponds instead to the neutral command around which the motion can be slowed down or interrupted.

In both programming branches, the opposite-arm bracelet is used to estimate the co-contraction level and to supervise transitions toward the dedicated training state. When the cocontraction signal is maintained above the threshold defined in Section 4.2, the FSM switches from State 1 to State 2 or from State 3 to State 4, depending on the currently active task phase. The successful state transition is notified through two short vibrotactile pulses followed by one long pulse. After an 8 s delay, three short pulses announce the start of gesture acquisition, and a 10 s long vibration at intermediate intensity instructs the operator to maintain the target gesture for the full recording window. The same vibrotactile sequence is repeated for all training cases. Once the new gesture samples have been collected and the incremental regression model has been updated, the FSM returns to the operative state and the user can continue programming the task component associated with the current phase.

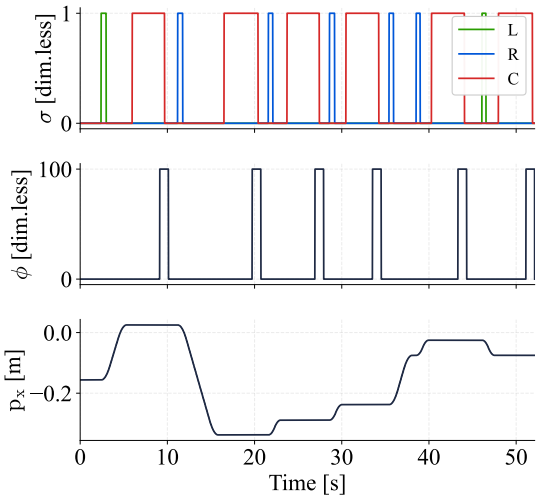


Figure 4.8 State 1: *Picking Sequence*. The decoded symbolic command σ alternates among *left*, *right*, and *confirm*; the vibrotactile signal ϕ marks the corresponding interaction events; and the robot position p_x evolves according to the selected sequence.

4.3.3 Experimental Evidence

The experimental protocol follows the two-stage programming logic described above. The operator first enters State 2, namely the training state associated with the *Picking Sequence* phase. The corresponding data acquisition is reported in Fig. 4.7. The upper panel shows the multichannel sEMG activity acquired during the prompted gesture trials. The second panel shows the cocontraction variable γ, whose crossing of the threshold activates the retraining procedure. The third panel reports the vibrotactile command level ϕ: two short pulses followed by one long pulse confirm the successful transition to the training state; after 8 s, three short pulses announce the beginning of gesture recording; and a subsequent 10 s stimulation at intermediate intensity identifies the interval in which the target gesture must be maintained. The bottom panel shows the FSM transition from State 1 to State 2 and then back to State 1. Overall, the figure highlights that muscular activity is concentrated within the prompted acquisition windows and that the training interaction is temporally coordinated by the cocontraction trigger and the vibrotactile feedback logic.

After the model update, the operator returns to State 1 and uses the learned discrete commands to define the picking sequence online. This interaction is illustrated in Fig. 4.8. The top panel reports the decoded symbolic outputs σ associated with the commands *left*, *right*, and *confirm*. The middle panel shows the corresponding vibrotactile notifications ϕ, which mark the discrete programming events. The bottom panel reports the resulting

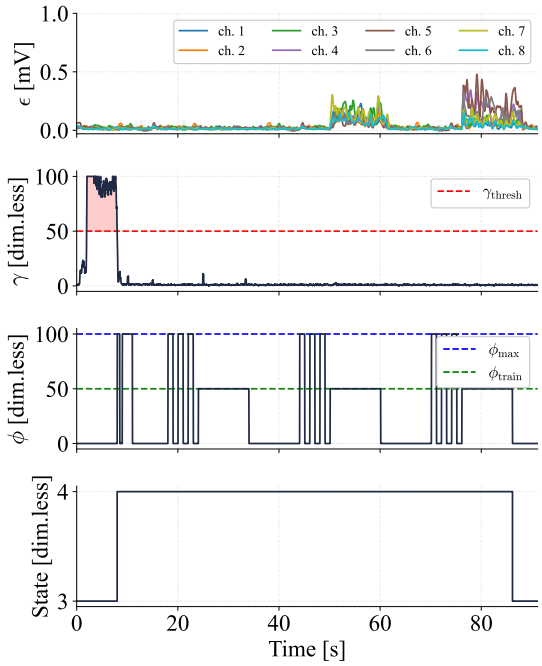


Figure 4.9 State 4: *Performing Trajectory Training*. The cocontraction level γ triggers the transition from State 3 to State 4; the vibrotactile signal ϕ provides the state-change confirmation, the warning pulses before acquisition, and the 10 s intermediate-intensity stimulation used to maintain each recorded gesture; and the sEMG traces e capture the three reference conditions used to update the continuous trajectory decoder before returning to State 3.

evolution of the programmed robot position p_x. The temporal alignment between the symbolic outputs and the piecewise changes of p_x shows that the robot reacts coherently to the discrete neuromuscular commands, allowing the user to progressively compose and validate the picking sequence.

The second part of the protocol concerns the *Performing Trajectory* phase. The related training session is performed in State 4 and is shown in Fig. 4.9. Also in this case, the cocontraction signal γ is used to enter the dedicated training state, and the same vibrotactile sequence is repeated for all recorded conditions *rest*, *open*, and *close*: two short pulses and one long pulse confirm the state transition, after 8 s three short pulses announce the beginning of recording, and a 10 s stimulus at intermediate intensity instructs the operator to maintain the required gesture. The multichannel sEMG traces exhibit distinct activation bursts during the prompted trials, and the bottom panel shows the transition from State 3 to State 4 and back to State 3, confirming that the training data are collected in well-delimited temporal windows and under explicit FSM supervision.

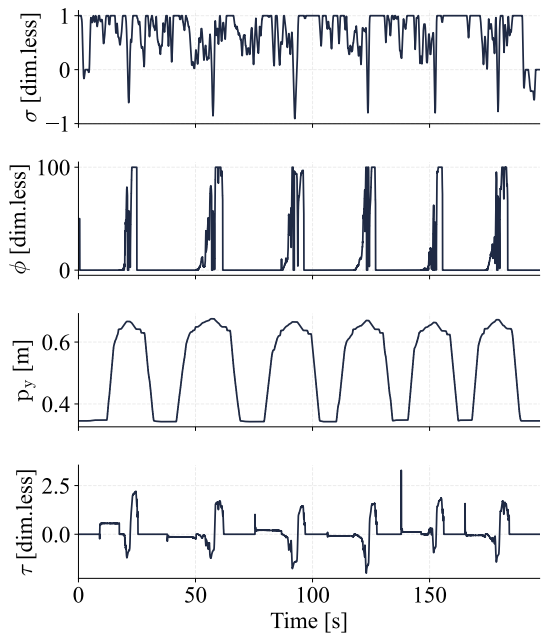


Figure 4.10 State 3: *Performing Trajectory*. The continuous decoded output σ modulates the robot progression along p_y, the vibrotactile command ϕ remains aligned with the interaction phases, and τ reports the contact-related response observed during the execution of the programmed motion.

Once the continuous decoder has been updated, the operator returns to State 3 and can modulate the robot motion online, as shown in Fig. 4.10. The decoded signal σ evolves continuously around its neutral condition and drives the commanded progression of the robot along the task coordinate p_y. The vibrotactile output ϕ remains synchronized with the interaction phases, while the variable τ reflects the contact-related response observed during the motion. The repeated forward and backward modulations of p_y, together with the corresponding evolution of σ, indicate that the user can shape the trajectory in a progressive manner before handing the task back to the autonomous controller. The robot motion execution and the vision- and tactile-based pin insertion stage that follow the HITL programming phase are consistent with the connector-assembly strategy described in (Caporali et al. (2026)). Combined with the previously defined *Picking Sequence*, this procedure yields the complete robot program used to execute the autonomous cable insertion reported in Fig. 4.6.

Figure 4.11 reports aggregated performance indicators for the robotic connector assembly case study evaluated on a cohort of 10 subjects. The figure summarizes the average completion time for the full programming procedure, including the HITL definition of the picking sequence and trajectory together with the subsequent task execution, as well as the corre-

sponding autonomous success rate achieved during cable insertion. The reported average completion time is 861.7 s, while the autonomous success rate is 80.0%. These results show that the proposed HITL interface allows the operator to program the task end-to-end and that, once the insertion behavior has been specified, the robotic system can autonomously execute the overall connector assembly procedure in the tested case study.

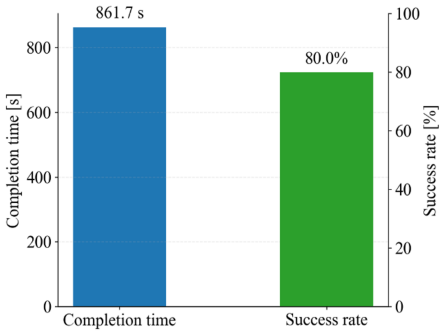


Figure 4.11 Aggregated performance indicators over 10 subjects. Average completion time for the full robotic connector assembly programming procedure, including HITL programming and execution, together with the corresponding autonomous task success rate during cable insertion.

4.4 Discussion and Thesis-Level Implications

This chapter presented a modular Human-in-the-Loop design pattern for robotic manipulation programming in industrial settings through neuromuscular interfaces. The proposed framework explicitly separates the fundamental interaction loop, the intent decoding module, the sensory feedback logic, and the supervisory finite-state machine that coordinates the different programming phases. This organization makes it possible to structure robot programming as a sequence of reusable HITL modules, while preserving the possibility of incremental adaptation through online retraining and state-dependent interaction policies.

The proposed methodology was instantiated in a robotic connector assembly case study, in which the operator programmed both a discrete *Picking Sequence* phase and a continuous *Performing Trajectory* phase through sEMG-based interaction. The experimental results showed that the architecture supports coherent transitions between operative and training states, allows the user to define the task end-to-end, and enables subsequent autonomous execution of the programmed manipulation. The aggregated evaluation on 10 subjects further

reported an average completion time of 861.7 s and an autonomous success rate of 80.0%, providing proof-of-concept evidence for structured industrial programming scenarios.

Overall, the results indicate that the proposed design pattern can reduce the gap between direct human interaction and structured robot programming, while retaining modularity, reconfigurability, and robustness to signal non-stationarity. The present study does not include a baseline programming interface, an ablation of the architectural modules, or a standardized usability comparison, so the findings should be interpreted as proof-of-concept rather than as broad usability validation. Future developments will address the extension of the framework to more complex multi-step assembly procedures, the integration of additional multimodal interaction channels, and the systematic evaluation of usability, generalization, and long-term adaptation across users and tasks. Within the overall thesis, this chapter therefore adds the interaction-architecture layer missing between system-level manipulation and later probabilistic control contributions. It shows that, under industrial interaction uncertainty, not only sensing and control but also task programming itself benefits from modularization, explicit supervision, and bidirectional human–robot communication.

Chapter 5

Probabilistic Contact Phase Inference for Force Control

5.1 Force Regulation under Human-Interaction Uncertainty

Grasping everyday objects –for example, picking up a fruit from a grocery shelf– requires accurate grip-strength regulation. If the applied force is too high, the object can be damaged; if it is too low, slippage may occur and the task fails. This requirement extends far beyond food handling. As the global population ages, elderly-care applications are becoming increasingly relevant, and service robots are expected to interact safely with users in daily-life routines. In these scenarios, robots may assist during motion support, feeding, and object handling. While humans naturally perform these tasks with high dexterity, robotic hands still present clear limitations compared with the human hand, making real-time force modulation a central control problem. Several studies have addressed this issue through analytical formulations based on precomputed contact models (Zaidi et al. (2017)). Although effective in structured environments, these methods can degrade when object properties are only partially known or vary over time (e.g., shape, size, weight, compliance), and when uncertainty affects both the human and robotic sides of the loop. In this context, sensor-driven control represents a practical strategy for coping with human, robot, and environmental variability. As reported in (Osborn et al. (2016); Thomas et al. (2023)), integrating feedback is essential for adaptive grip-force regulation. A number of haptic frameworks have been proposed (Ficuciello et al. (2021); Selvaggio et al. (2019)), but many of them mainly assist the reaching-to-grasp stage. Other approaches provide passive stabilization support (Zhuang et al. (2019)), or infer intended force from surface electromyographic (sEMG) signals (Wen et al. (2020)). Nevertheless, many supervised solutions still suffer from long preparation procedures and

limited generalization. In addition, pure feedforward force-control schemes are intrinsically less robust to real-world variability. To overcome these limitations, grip-strength regulation for human-in-the-loop (HITL) robot-hand control is formulated here as an unsupervised and probabilistic feedback-driven problem. In the proposed framework, tactile measurements are used to extract probabilistic grasp phases through Hidden Markov Models (HMMs), while sEMG signals provide user intent for motion control. Phase probabilities are then exploited to modulate the mapping between user command and finger-joint velocity variation, enabling fine, smooth, and controllable force regulation. The reference task is representative of everyday operations: the robotic hand must grasp an object and progressively increase force until a desired level is reached, typically through precision grasping, where contact is dominated by the fingertips rather than the palm. The main contributions of this chapter can be summarized as follows:

- The use of HMMs to probabilistically model HITL grasping actions as stochastic process encoding of tactile data.
- The development of a shared-autonomy framework exploiting HMM-based probabilistic information related to the different phases of the grasping process to modulate the level of shared-autonomy for fine, smooth, and controllable grip strength regulation.

Two controlled experiments with 10 able-bodied subjects were first conducted to systematically evaluate the proposed strategy. Hardware transferability was then investigated on both an anthropomorphic hand and an industrial gripper, using tripodal and parallel grasp configurations, respectively. A further realistic validation was performed with an additional group of 10 able-bodied subjects and one participant with upper-limb loss. The resulting evidence supports smoother and more controllable grip-strength regulation than a control framework without the proposed shared-autonomy mechanism, within the tested tasks and hardware settings. From the thesis perspective, this chapter addresses uncertainty during contact-rich manipulation by combining tactile sensing, probabilistic phase inference, and adaptive gain modulation. Its main limitation is that the user study remains relatively small and the evidence with limb loss is pilot in nature.

5.2 State-of-the-Art Context for Probabilistic Shared Autonomy

The mechanical complexity of modern robotic hands, combined with uncertainty during contact-rich interaction, makes human-centered manipulation intrinsically challenging. For this reason, many studies have explored strategies that assist users through shared-autonomy, namely by combining human commands with robot-side intelligence to improve overall task performance. For instance, (Ficuciello et al. (2021)) proposed a kinematic mapping for robot-assisted surgical manipulation, aimed at reducing operator-side complexity while preserving controllability. However, kinematic alignment alone is typically insufficient to guarantee robust manipulation. Dynamic aspects such as grip-force regulation and grasp stability remain fundamental, especially when handling fragile or deformable objects. Classical model-based force-control methods generally rely on precomputed contact/object models (Zaidi et al. (2017)). While these approaches can work well in controlled scenarios, they often degrade in realistic settings due to changes in object properties, environmental disturbances, and unforeseen interaction events. In these conditions, sensor feedback can significantly improve adaptability. The quality and reliability of feedback delivered to the user is therefore a key design factor. Prior work has already shown the benefits of haptic feedback in myoelectric hand control (Thomas et al. (2023, 2019)). In (Motamedi et al. (2016)), haptic cues were found to be advantageous over visual feedback in slip detection, contact detection, and grasp control tasks. Haptic guidance has also been integrated in several telemanipulation frameworks (Adjigble et al. (2019); Selvaggio et al. (2019)), (Abi-Farraj et al. (2019)). For example, (Selvaggio et al. (2019)) used impedance guidance to improve grasp execution in suturing, while (Adjigble et al. (2019)) and (Abi-Farraj et al. (2019)) focused on assisted grasping in cluttered environments. However, many of these methods mainly support the reaching-to-grasp phase and do not explicitly model force regulation during sustained object interaction. A different direction was proposed in (Zhuang et al. (2019)), where shared control was used to increase contact area and improve robustness, though without directly addressing fine grip-force modulation. Nonetheless, real-time force regulation remains essential when object characteristics and interaction conditions are uncertain (Regoli et al. (2016)). Among available feedback modalities, vibrotactile stimulation is widely used due to its non-intrusiveness, ease of integration, and practical effectiveness (Thomas et al. (2023)). Previous studies reported improved stiffness discrimination (Witteveen et al. (2013)), shorter manipulation times (Raveh et al. (2018)), and better grip-force tuning (Thomas et al. (2023)). In parallel, several works estimated intended force directly from sEMG signals, using steady-

state, transient, or full-envelope features with machine-learning methods (Martinez et al. (2020a,b)), (Wen et al. (2020)). For example, (Wen et al. (2020)) employed neural-network regression within an admittance framework to infer desired force from sEMG.

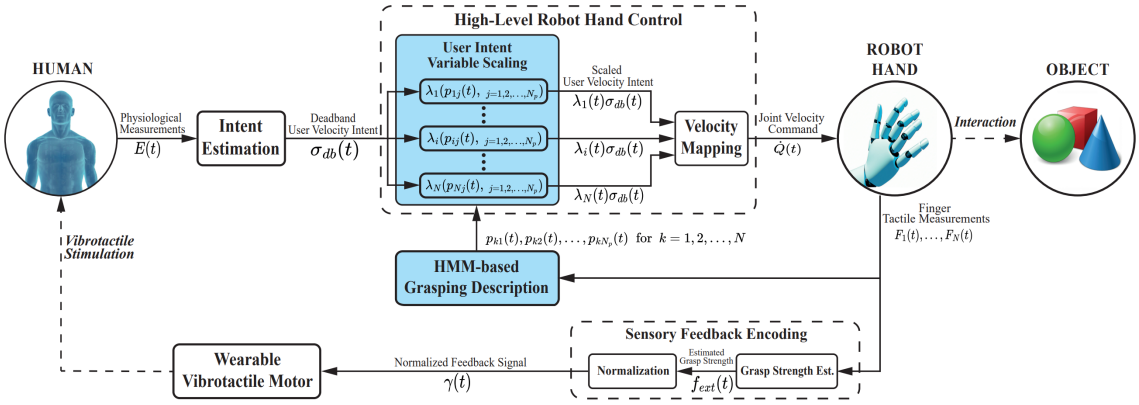


Figure 5.1 Proposed shared-autonomy framework for grip strength regulation in HITL robotic hand control applications. The user velocity intent $\sigma_{\mathrm{db}}(t)$, extracted from physiological measurements (the sEMG signals $E(t)$), is scaled, for each finger, through phase-dependent gains $\lambda_i\big(p_{ij}(t), j = 1,2,\ldots,N_P\big)$ for $i = 1,2,\ldots,N$. These gain terms are a function of the finger grasping process phase probabilities $p_{ij}(t)$, $j = 1,2,\ldots,N_P$, which are computed from finger tactile measurements $F_i(t)$ for $i = 1,2,\ldots,N$ via an HMM-based grasping description phase (according to (2) and (3)). The resulting scaled intents are mapped to joint velocity commands $Q(t)$ to operate the robotic hand. Finally, finger tactile measurements are also used to estimate grip strength $f_{\mathrm{ext}}(t)$, which is then encoded and conveyed back to the user via a vibrotactile motor.

In that line of work, force sensors are often required to generate labels, and substantial supervised training is needed. Similarly, (Fang et al. (2022)) proposed a convolutional architecture for simultaneous hand-gesture and force-level recognition. Although this can improve usability in prosthetic systems, it usually constrains users to predefined discrete force classes. In contrast, this chapter addresses continuous grip-strength regulation in uncertain environments through a probabilistic shared-autonomy framework based on HMMs, which are known to handle observation uncertainty effectively (Bekiroglu et al. (2011)). HMMs have already been used in robot-hand contexts: (Bernardin et al. (2005)) applied them to recognize grasp sequences from glove and tactile signals, and (Son (2022)) modeled successful grasp inference in clutter through an HMM with particle filtering from video data. However, these approaches did not directly target online grip-force regulation, which is critical in daily-life manipulation. Here, HMM-based tactile phase probabilities are exploited

to continuously adapt shared-autonomy and obtain smooth, fine, and controllable force modulation.

5.3 Probabilistic Phase Modeling Framework

This section details the HMM-based shared-autonomy architecture shown in Fig. 5.1. Here, shared-autonomy denotes a control paradigm in which robot assistance is continuously integrated with user commands during task execution (Dragan and Srinivasa (2013a)). In this application, assistance is implemented through an adaptive command mapping: the high-level user velocity input is continuously transformed into finger-joint velocity variations with phase-dependent scaling, enabling smoother and more effective HITL grasp execution. This formulation is closer to policy blending (Dragan and Srinivasa (2013a)) than to hard autonomy scheduling, because human and robot contributions are blended continuously rather than alternated. The HMM-based grasping-description block extracts probabilistic grasp phases from tactile measurements (one model per finger). These phase probabilities are then used by the high-level robot-hand control block to modulate user intent, extracted in parallel from the intent-estimation block. Finally, the sensory-feedback encoding block converts grip information into vibrotactile cues for the operator. Let the physiological input be denoted by $E(t) \in \mathbb{R}^{I_c}$, where I_c is the number of acquisition channels. The controlled device is a robotic hand with N fingers; each finger has joint-velocity vector $q_i(t) \in \mathbb{R}^{N_{\text{joints}}}$, with $i \in \{1,2,\ldots,N\}$ and N_{joints} joints per finger. For formulation clarity, the method is first described for a single finger and then extended to the full hand. Tactile feedback for finger i is denoted by $F_i(t) \in \mathbb{R}^{F_c}$, where F_c is the number of tactile channels (assumed equal across fingers for simplicity).

5.3.1 HMM-based Grasping Process Description

Fig. 5.2 summarizes the design intuition of the proposed framework. The grasping process –including reaching-to-grasp and contact regulation– is represented, for each finger, as a sequence of latent interaction phases inferred from tactile data through an HMM. Each phase is associated with a specific control action through a corresponding gain value. The resulting shared-autonomy policy modulates the user contribution to finger-joint velocity according to the online phase probabilities. In our architecture, each finger is modeled independently by its own HMM trained on that finger's tactile measurements. This ensures methodological consistency across fingers while allowing functional differences to emerge from data. For

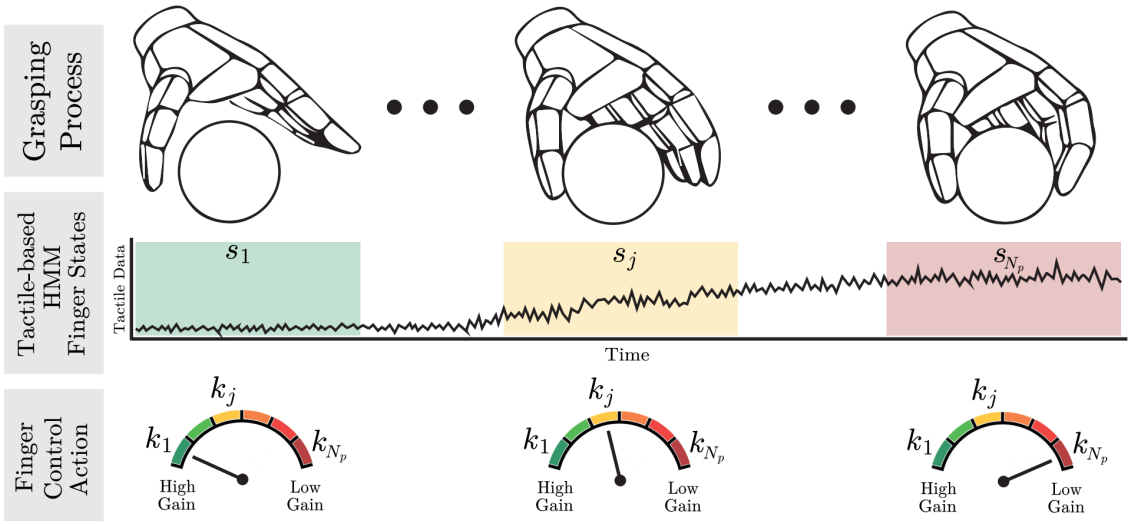


Figure 5.2 Shared autonomy design intuition: the grasping process can be decomposed by each finger into distinct grasping process phases $s_1,\ldots,s_j,\ldots,s_{N_p}$, which are predicted from tactile data through a properly trained HMM. These phases, in turn, are associated with different finger control actions through the associated gain terms $k_1,\ldots,k_j,\ldots,k_{N_p}$, which vary the user's contribution to finger joint velocity variations.

notation clarity, the finger index is omitted in this subsection, and the formulation is written for one generic finger. Let $S=\{s_1,s_2,\ldots,s_{N_p}\}$ be the set of hidden grasp phases, and let $F(t)\in\mathbb{R}^{F_c}$ denote the continuous multidimensional tactile observation. We assume that each state s_j, $j\in\{1,2,\ldots,N_p\}$, emits observations according to a Gaussian density $b_j(F(t))$:

$$b_j(F(t)) = \frac{1}{(2\pi)^{\frac{F_c}{2}}|\Sigma_j|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(F(t)-\mu_j)^T\Sigma_j^{-1}(F(t)-\mu_j)\right) \tag{5.1}$$

where μ_j and Σ_j specify the mean vector and covariance matrix of the multivariate normal observation distribution for state s_j. Although richer emission models (e.g., Gaussian mixtures) can approximate tactile distributions more precisely, our objective is to capture robust phase-level structure rather than full tactile complexity. Following (Asfour et al. (2008); Calinon et al. (2010)), multivariate-Gaussian HMMs provide a suitable balance between expressiveness and controllability. The grasp evolution of each finger is therefore represented by an HMM $\Theta=(A,B,\Pi)$ with parameters:

- The transition matrix $A = \{a_{jk}\} \in \mathbb{R}^{N_p \times N_p}$, with a_{jk} the probability of moving from state s_j to s_k.
- The observation parameters $B = \{(\mu_j, \Sigma_j)\}_{j=1}^{N_p}$.
- The initial probability distribution $\Pi = \{\pi_j\}_{j=1}^{N_p}$, with π_j the probability of starting in state s_j.

Here, we train the HMM to learn the parameters of the finger model that best describes the time evolution of tactile measurements. Specifically, the Expectation-Maximization (EM) algorithm (Moon (1996)) is applied on a properly constructed training dataset (Section 5.4.2) to find the maximum-likelihood estimate of model parameters $\hat{\Theta} = (\hat{A}, \hat{B}, \hat{\Pi})$. To this aim, the transition probability matrix A and the observation parameters B are randomly initialized, with the only requirement that the grasping process be described by sequential states (i.e., only states close to each other have non-zero transition probabilities), reflecting the inherent sequential nature of the task. Similarly, the initial state distribution Π is defined in such a way that the grasping process always starts in the first hidden state of the sequence, a condition that can be easily be met with a proper design of the HMM training phase, for example by starting the recording with the robotic hand open. Once trained, the model can be used online to infer hidden-state probabilities from new observations. Specifically, given the estimated parameter set $\hat{\Theta} = (\hat{A}, \hat{B}, \hat{\Pi})$ and a new window of tactile measurements $F_{\text{online}} \in \mathbb{R}^{F_c \times T_{\text{online}}}$, we are interested in performing posterior decoding for each state in the model at each time instant:

$$P\big(z(t) = s_j \mid F_{\text{online}}, \hat{\Theta}\big) = \frac{P\big(z(t) = s_j, F_{\text{online}} \mid \hat{\Theta}\big)}{P\big(F_{\text{online}} \mid \hat{\Theta}\big)} \tag{5.2}$$

where $z(t) \in S$ indicates the state of the system at time t. Within this framework, we can define the probability of being in state s_j at time t as $p_j(t) = P(z(t) = s_j)$, with $j \in \{1, 2, \ldots, N_p\}$. Since we consider the interaction between the robotic hand and everyday objects, viscoelastic contacts often occur, which can be modeled as spring-damper systems. According to this model, the grip strength increases proportionally to the deformation of the fingertip-object system. Therefore, given the same user input, if we reduce the effect of this input on the movement of the robotic hand, we can achieve more precise control of the grip strength. Our framework achieves this by using state probabilities to adjust the mapping between user input and finger joint velocity variations by varying the shared-autonomy gain $\lambda(t)$, given by:

$$\lambda(t) = \sum_{j=1}^{N_p} k_j p_j(t) \tag{5.3}$$

Notably, we chose a linear combination of state probabilities to guarantee smooth and continuous modulation of user input throughout the grasping process, thus ensuring predictable behavior of the robotic hand. Indeed, as the state probabilities sum to one, the gain function turns out to be a convex combination of the parameters $k_1, k_2, \ldots, k_{N_p}$. Consequently, the resulting shared-autonomy gain is always between the minimum and maximum gain values, and transitions between them occur smoothly. Furthermore, the parameters are assigned to shape the behavior of the robotic hand while preserving fine manipulation capabilities. In particular, our design approach is based on the definition of the maximum $k_{\max}$ and minimum $k_{\min}$ gains, regardless of the number of hidden states, whereas the intermediate gains are evenly spaced between the two extreme values. These two gains are designed to meet two important requirements of modern telemanipulation applications: reactivity and touch sensitivity, which have been proven to be paramount for seamless and accurate robotic hand control (Engeberg and Meek (2008); Milstein et al. (2018)). As highlighted in the literature, temporal coherence between user input and robot movements is essential for intuitive interaction (Farina et al. (2014)). To maximize temporal coherence between the two hands, we computed the maximum gain as:

$$k_{\max} = \arg\min_{k} |T_h - T_r(k)| \tag{5.4}$$

where T_h and T_r are the human and robotic hand closure time, defined as the time required to go from the fully open to the fully closed configuration (similar to the grasp cycle time defined in (Falco et al. (2020))). In this specific application, given a ramp velocity input profile (the operator control command is a ramp that goes from 0 to a maximum value $\sigma_{\max}$ in a certain time interval, which corresponds to the human hand moving from the open configuration to the closed configuration), the maximum gain was computed as:

$$k_{\max} = \frac{2\Delta q}{S_G \sigma_{\max} T_h} \tag{5.5}$$

where Δq is the space covered by the robotic fingers during the closing movement, and S_G is the grasp synergy matrix (which will be explained in the next section). On the other hand, the minimum gain is designed to ensure delicate grasping when contact occurs between the hand and the object. Therefore, the minimum gain was computed to maximize touch sensitivity (defined in (Falco et al. (2020)) as the smallest contact force exerted by a robotic finger on an object). To this aim, following the same evaluation procedure, the robotic hand was commanded to close on an object at different joint velocities $q = kS_G\sigma_{\max}$, and the corresponding peak force $f_{\text{ext}}^{\text{peak}}$ at the time of first contact with the object was measured.

Finally, $k_{\min}$ was computed as the maximum gain value that keeps the peak force below a certain sensor-specific threshold f_{th}, which was chosen according to sensor resolution:

$$k_{\min} = \max \left\{ k \mid f_{\text{ext}}^{\text{peak}} \leq f_{\text{th}} \right\} \tag{5.6}$$

This design is based on two complementary requirements that are fundamental in human-driven manipulation tasks. If the low gain (i.e., the one obtained by satisfying the touch-sensitivity requirement) is used in all states, closure time degrades. Conversely, using the high gain in all phases degrades touch sensitivity. The latter case is therefore used as a baseline in the experiments that follow. These considerations are qualitatively supported by the gain analysis presented in Section 5.5.4, where the effect of different values of $k_{\max}$ and $k_{\min}$ on system performance is examined. A key aspect of the HMM-based framework presented in this section is sensor-type independence, which is important for portability. HMMs do not require sensor-specific feature engineering to model the temporal evolution of the grasping process. By working directly on raw sensor data, the same probabilistic structure can in principle capture grasping dynamics across sensing technologies, transmission frequencies, and sensitive areas.

5.3.2 High-Level Robot Hand Control

In this stage, the per-finger gain terms are used to continuously regulate how control authority is shared between operator and robotic device. We adopt a synergy-based formulation so that all finger joints are controlled through a compact input representation while preserving coordinated motion. The control law is:

$$\dot{Q}(t) = \Lambda(t) S_G \sigma_{\text{db}}(t) \tag{5.7}$$

where $\dot{Q}(t) = [\dot{q}_1^T(t)\ \dot{q}_2^T(t)\ \dots\ \dot{q}_N^T(t)]^T \in \mathbb{R}^{N \cdot N_{\text{joints}} \times 1}$ is the hand joint velocity command, $\sigma_{\text{db}}(t) \in \mathbb{R}$ is the deadband user velocity intent derived from human physiological measurements (Section 5.3.4), while $S_G = [S_{G1}^T\ S_{G2}^T\ \dots\ S_{GN}^T]^T \in \mathbb{R}^{N \cdot N_{\text{joints}} \times 1}$ is the hand grasp synergy matrix used to reduce the dimensionality of the control space, with finger synergies $S_{Gi} \in \mathbb{R}^{N_{\text{joints}} \times 1}$, $i \in \{1, 2, \dots, N\}$. These matrices are computed by taking the difference between the extended and closed finger configurations. Finally, the diagonal gain matrix is

defined as

$$\Lambda(t) = \operatorname{diag}\big(\underbrace{\lambda_1(t),\ldots,\lambda_1(t)}_{N_{\text{joints}}\text{ times}},\ldots,\underbrace{\lambda_N(t),\ldots,\lambda_N(t)}_{N_{\text{joints}}\text{ times}}\big) \in \mathbb{R}^{N\cdot N_{\text{joints}}\times N\cdot N_{\text{joints}}}, \tag{5.8}$$

which contains, for each finger, the shared-autonomy gain term computed as in (3). Although (7) adopts fixed synergy matrices –which could, in principle, limit the adaptability of the robotic hand– our control framework, being independent for each finger, effectively adapts the shape of the robotic hand to the grasped object. The finger-specific shared-autonomy gains dynamically modulate each synergistic contribution, enabling real-time grasp adaptation. This behavior parallels the soft-synergy model of (Bicchi et al. (2011)), although in our case adaptation occurs at the control level rather than via mechanical compliance.

5.3.3 Closed-Loop Regularity and Boundedness

Rapid tactile fluctuations may cause abrupt variations in estimated grasp-phase probabilities. To prevent discontinuities in gain modulation, we apply a first-order exponential filter to probability signals. The filter is defined as:

$$p_{j,\text{filt}}(n) = \alpha p_j(n) + (1-\alpha)p_{j,\text{filt}}(n-1) \tag{5.9}$$

where $p_j(n)$ and $p_{j,\text{filt}}(n)$ are the input and filtered grasp-phase probability signals at sampling time n, respectively, and the coefficient α was selected to attenuate only high-frequency components above the working bandwidth of the HMM probability dynamics under nominal tactile sensing conditions.

This filtering step does not provide a formal Lyapunov stability proof for the complete human–robot–object system. It does, however, support a boundedness and regularity argument under explicit assumptions. Since $p_j(t) \in [0,1]$ and $\sum_j p_j(t) = 1$, the shared-autonomy gain $\lambda(t) = \sum_j k_j p_j(t)$ is a convex combination of $\{k_j\}$, hence $k_{\min} \le \lambda(t) \le k_{\max}$ for all t. With bounded user intent $|\sigma_{\text{db}}(t)| \le \bar{\sigma}$, the commanded velocity satisfies $\|\dot{Q}(t)\| \le \|\Lambda(t)\|\|S_G\|\bar{\sigma} \le k_{\max}\|S_G\|\bar{\sigma}$, which guarantees bounded control action. In addition, because the probability filter is first order and the gain depends affinely on the filtered probabilities, abrupt switching in the gain profile is avoided and the commanded motion remains piecewise smooth.

Under the additional assumptions that the inner robotic-hand servo is well tuned, joint friction and actuator limits keep the low-level motion bounded in the operating range, and the

grasped object behaves passively over the considered interaction regime, these properties are consistent with the smooth closed-loop behavior observed in the reported experiments. The claim made here is therefore limited to smooth gain modulation and bounded commanded motion under the stated assumptions, not to a general proof of closed-loop stability for arbitrary contacts, users, or objects.

5.3.4 Feed-Forward and Feedback Interfaces

This subsection presents the feed-forward and feedback interfaces that connect the shared-autonomy core with the human user. Specifically, it details the implementation adopted for the intent-estimation and sensory-feedback encoding blocks in Fig. 5.1.

Intent Estimation

Robot-hand opening and closing are controlled through a velocity input extracted from physiological measurements. In this implementation, the input is forearm sEMG. Let $E(t) \in \mathbb{R}^8$ be the RMS vector of the 8-channel signal. According to (Jiang et al. (2008)), synergistic muscles share latent activation components (neural drives), so RMS measurements can be approximated as a linear combination of such components. Although linear synergies simplify the full neuromuscular process, they are consistent with established proportional myoelectric control pipelines based on linear models and NMF (Meattini et al. (2018); Nowak et al. (2022)). Since our primary objective is robust grip-force modulation rather than multi-class gesture decoding, this representation offers a practical and predictable control variable. Under antagonistic actuation assumptions, hand opening/closing is represented as a single DoF produced by flexor and extensor groups (Meattini et al. (2018)). Hence, $E(t)$ is written as:

$$E(t) = S_M U(t) = \begin{bmatrix} s_{Me} & s_{Mf} \end{bmatrix} \begin{bmatrix} u_e(t) \\ u_f(t) \end{bmatrix} \tag{5.10}$$

where $s_{Me}, s_{Mf} \in \mathbb{R}^8$ are the extensors and flexors components of the muscular synergy matrix $S_M \in \mathbb{R}^{8\times 2}$ and $u_e(t), u_f(t) \in \mathbb{R}$ are the corresponding activation signals. An estimation of the matrix S_M is needed to extract neural commands from sEMG to control the robotic hand. This can be achieved through a short calibration phase, after which the Non-negative Matrix Factorization (NMF) algorithm is applied to the recorded sEMG signals following the semi-unsupervised approach outlined in (Meattini et al. (2018)). To be precise, let us denote by $E_{\text{train}} \in \mathbb{R}^{8\times T}$ the RMS value of the sEMG signal recorded during a training phase in which the user has to perform two opening/closing motions of the hand (T is the recording length).

This signal matrix is factorized into the product of two non-negative matrices $E_{\text{train}} = S_M U_{\text{train}}$ via the NMF algorithm. After that, the pseudo-inverse of matrix S_M, denoted as S_M^+, is used to online extract neural commands from sEMG signals, as $U(t) = S_M^+ E(t)$. To address the problem of non-unique solutions given by the NMF algorithm, a check on the sign of the output is performed, as it is the only indeterminacy in the solution (Meattini et al. (2018)). A further step is performed to get a single velocity closure command $\sigma(t) \in \mathbb{R}$ from the two antagonist signals:

$$\sigma(t) = a\big(u_e(t) - u_f(t) + b\big) \tag{5.11}$$

where a and b are the multiplicative and additive factors used to remap the signal into the interval $[-0.5, +0.5]$ (to normalize the overall signal excursion). This pre-experiment calibration procedure is essential to address intersubject variability. Indeed, due to physiological differences, sEMG signals vary significantly from one subject to another. Therefore, as in common myoelectric control practice (e.g., (Jiang et al. (2008, 2013))), our framework extracts a subject-specific muscular synergy matrix S_M, which captures individual's muscle activation patterns. The neural commands derived from S_M are then normalized into a single velocity input signal, effectively mapping sEMG signals into a single open/close command of the robotic hand. This procedure guarantees a similar device behavior across users, thereby obtaining comparable performance. Lastly, the input $\sigma(t)$ is further manipulated to allow the user to smoothly maintain the desired pose while muscles are not contracted, via the introduction of a deadband. Thus, we can define the deadband user velocity intent $\sigma_{\text{db}}(t)$ as:

$$\sigma_{\text{db}}(t) = \begin{cases} \sigma(t) - \bar{\sigma}_{\text{inf}}, & \sigma(t) \leq \bar{\sigma}_{\text{inf}} \\ 0, & \bar{\sigma}_{\text{inf}} < \sigma(t) < \bar{\sigma}_{\text{sup}} \\ \sigma(t) - \bar{\sigma}_{\text{sup}}, & \sigma(t) \geq \bar{\sigma}_{\text{sup}} \end{cases} \tag{5.12}$$

where $\bar{\sigma}_{\text{inf}}$ and $\bar{\sigma}_{\text{sup}}$ are two thresholds used to shape the deadband region according to user preferences. Hence, whenever the closure signal falls between these two values (i.e., when muscles are at rest), the robotic hand keeps its position since the deadband user velocity intent is zero.

Sensory Feedback Encoding

Motivated by the positive evidence on vibrotactile feedback for force control, the proposed architecture delivers grip-related cues through vibrotactile stimulation on the upper arm. We deliberately avoid direct automatic stiffness modulation from slip or object estimators

in order to preserve hardware independence and reduce dependence on rigid precomputed models that may fail in uncertain conditions. In this chapter, grip strength is defined as an estimate of internal grasp forces, i.e., contact forces with zero net resultant on the object. The exact computation depends on hand mechanics, but practical approximations can be obtained from geometric considerations on contact configuration (Section 5.4.1). Direct use of raw sensor values would limit portability because sensing technologies differ in principle, range, and sampling. To overcome this issue, grip strength is represented as the norm of tactile measurements $F(t)$ and mapped to a normalized feedback signal $\gamma(t) \in [0,1]$, which modulates vibration ON-time through PWM. This yields a compact scalar encoding that supports transfer across heterogeneous sensors. Provided that sensor outputs increase monotonically with applied force, the norm gives a practical, though approximate, estimate of grip strength.

5.4 Experimental Protocol and Statistical Validation

The goal of our experiments was to validate the proposed shared-autonomy framework and assess users' ability to finely regulate grip strength in HITL robotic hand control applications. Research has already extensively investigated the effectiveness and benefits of haptic feedback for myoelectric control, and in particular, the importance of vibrotactile stimulation for appropriate grip strength regulation. In this context, the feedforward and feedback blocks are considered complementary and indivisible parts of the control loop in our experiments, and their synergistic performances are investigated in the following.

5.4.1 Experiment Setup

The framework was implemented in Python, using the `hmmlearn` library for HMM training/inference and the Robot Operating System (ROS) for communication and synchronization across software blocks.

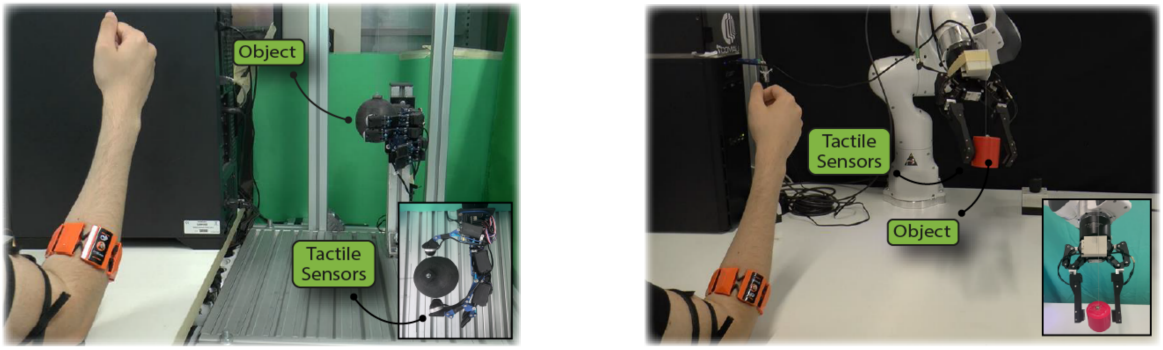
Subjects

Ten able-bodied participants (aged 30 ± 4.0 years, 7 males and 3 females) took part in the first two experiments. A second group of ten able-bodied participants (aged 28.2 ± 5 years, 7 males and 3 females), together with one participant with transradial limb loss, was involved in the third experiment. Experimental activities were performed at an Italian prosthetic center (approval ID 84-2025-SPER-AUSLBO). Informed consent was collected before each session

in accordance with the Declaration of Helsinki, along with a detailed explanation of the protocol.

Robot Hands and Sensor Equipment

To assess hardware generalization, experiments were conducted with both an anthropomorphic hand and an industrial gripper. Anthropomorphic hand: the AR10 (Active8 Robots) is a commercial five-finger hand with 10 DoFs (2 per finger). In our tests, it executed tripodal grasps on objects of different properties. Three force sensors (OMD-20SE-40N, Optoforce) were mounted on thumb, index, and middle fingertips to measure contact forces along x, y, and z. During dataset generation and in the first experiment, the AR10 was mounted in the fixed setup shown in Fig. 5.3, panel (a), composed of grounded metal profiles to ensure repeatable conditions; the target object was suspended at a fixed location with a fishing line. Industrial gripper: the Robotiq 2F-85 adaptive gripper was instrumented with tactile pads made of a 5×5 optoelectronic taxel grid (Cirillo et al. (2021)). In all experiments, the gripper was mounted as end-effector of a 7-DoF Panda robot, as shown in Fig. 5.3, panel (b), and in Fig. 5.4, panel (b).



(a) (b)

Figure 5.3 Setup for the fixed hand-object experiment: (a) the AR10 robotic hand and (b) the industrial gripper. In both configurations, the user wears a myoelectric armband to control the robotic device with the proposed shared-autonomy strategy and receives vibrotactile feedback on the upper part of the same arm. Finger tactile sensors provide real-time measurements used for grip-strength estimation and phase prediction. To complete the task, the user was instructed to control the robotic hand via myoelectric signals, grasp the object (a plastic ball of diameter 6 cm for the AR10 and a plastic cylinder of diameter 3 cm for the industrial gripper), and reach a desired strength.

Myoelectric Signals Acquisition and Processing

Both the anthropomorphic hand and the industrial gripper were controlled through forearm sEMG signals. Raw data were acquired at 1 kHz with the OYMotion GForce-Pro wearable armband, equipped with 8 dry differential electrodes and low-power Bluetooth streaming. The armband was positioned about 5 cm below the elbow, approximately over the main muscles involved in hand opening and closing (Flexor Digitorum Superficialis and Extensor Digitorum Communis). The processing chain included a 50 Hz notch filter (power-line rejection), a 20 Hz high-pass filter (baseline drift attenuation), and RMS extraction over a 200 ms sliding window. The resulting features were provided to the intent-estimation block described in Section 5.3.4. A short calibration was performed before each experiment, during which users executed two opening and closing movements. This dataset was used to estimate subject-specific synergies and produce a normalized control input, improving cross-user consistency.

Wearable Vibrotactile Motor

A two-piece armband consisting of two vibration coin motors (Grove Vibration Motor Module, Seeed Technology Co., Ltd.; coin-type ERM DC motor, rated at 3 V and operating up to 5 V, providing a nominal vibration frequency of about 150 Hz at full drive, with integrated driver for PWM control) was developed to convey feedback information to the user; the wearable implementation is shown in Fig. 5.6, panel (b). Vibration motors were controlled at a frequency of 20 Hz via an Arduino platform using a PWM strategy with a period of 0.5 s. The duty cycle of the PWM was modulated by a normalized feedback signal $\gamma(t)$ in [0, 1], obtained as a function of the estimated grip strength $f_{\text{ext}}(t)$. As outlined in Section 5.3, the estimated grip strength was defined as a measure of the internal forces on the grasped object. For the AR10 hand, following the virtual finger model of (Iberall et al. (1986)), we can approximate the action of the index and middle fingers during tripodal grasp as the action of a single finger located in an intermediate position with respect to the two and opposing the thumb. In this scenario, the force exerted by the thumb can be approximated as equal and opposite to the force of the virtual finger. On the other hand, a two-finger grasp is performed with the industrial gripper. Therefore, the actions of the two fingers can be approximated as equal and opposite. We recognize that this assumption is the result of the specific hands used and their configuration. To obtain a more general solution, one could use an estimate of grip strength derived from all finger measurements, as proposed by (Cipriani et al. (2008)). However, the main objective here is to validate the

proposed shared-autonomy architecture. In this context, using the thumb and the gripper finger as references for grip strength is a practical simplification functional to the experimental validation of the framework. For this reason, tactile feedback measurements of the thumb and the gripper finger were used to estimate grip strength. Finally, to obtain a device-independent scalar value for vibrotactile feedback encoding, grip strength was represented as the norm of the measured signals. Given the force feedback measurements of the thumb finger, $F(t) = [f_x(t)\ f_y(t)\ f_z(t)]^T \in \mathbb{R}^3$ ($F_c = 3$ for each finger), the AR10 grip strength was estimated as:

$$f_{\text{ext}}(t) = \sqrt{f_x(t)^2 + f_y(t)^2 + f_z(t)^2} \in \mathbb{R} \tag{5.13}$$

Similarly, given the tension feedback measurements $F(t) = [f_1(t)\ f_2(t)\ \dots\ f_{25}(t)]^T \in \mathbb{R}^{25}$ ($F_c = 25$ for each finger), the gripper grip strength was calculated as:

$$f_{\text{ext}}(t) = \sqrt{\sum_{i=1}^{25} f_i(t)^2} \in \mathbb{R} \tag{5.14}$$

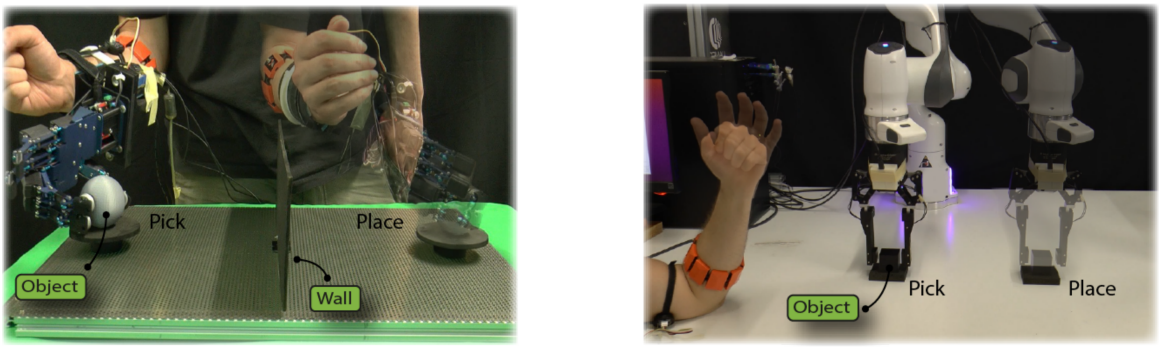
It is important to note that gripper feedback values are expressed in Volts, whereas AR10 feedback values are expressed in Newtons. While this mismatch may generate ambiguity, it does not affect our approach. Indeed, as explained in (Cirillo et al. (2021)), gripper voltage readings are directly related to the force applied to the pad. This is a crucial point toward the generalizability of our approach, which can be applied to different sensing technologies, as long as they provide information related to the force exerted on the object.

5.4.2 Experimental Protocol

The validation campaign included one HMM-training phase and three experimental evaluations. After hyperparameter optimization, we first performed a fixed hand-object experiment to compare four control models: constant gain, and shared-autonomy with 2, 3, and 4 hidden states. A second pick-and-place experiment was then carried out to assess transfer to different objects and dynamic interaction. Finally, a recipe-inspired sequential task was executed to evaluate force-regulation capability in a scenario closer to daily-life manipulation.

HMMs Training

A preliminary data-collection phase was designed to build the HMM training set and capture representative tactile dynamics during grasping. Ten hand-closure levels and four joint-velocity levels were selected according to actuator operating ranges, and each possible



(a) (b)

Figure 5.4 Setup for the pick-and-place experiment for (a) the AR10 robotic hand and (b) the industrial gripper. In both configurations, the user wears a myoelectric armband to control the robotic device with the proposed shared-autonomy strategy and receives vibrotactile feedback on the same arm. Finger tactile sensors provide real-time measurements used for grip-strength estimation and phase prediction. To complete the task, the user was instructed to perform pick-and-place of 9 objects of varying sizes and weights (3 cubes, 3 cylinders, and 3 spheres, each in three different dimensions) without exceeding a desired strength value.

combination was executed. Closure levels spanned the full hand range from fully open (0%) to fully closed (100%), with 10% increments, so as to include progressively higher interaction forces. In absolute units, this corresponded to 7900 (0%) to 2500 (100%) servo-position units for AR10, and 0 (0%) to 255 (100%) command units for the parallel gripper. Velocity levels were chosen as 500, 1200, 1900, and 3600 servo-position units/s for AR10, and 50, 100, 150, and 200 command units/s for the gripper. The resulting dataset consisted of 40 grasping trajectories per device. Although compact, this design is aligned with prior studies where HMMs were successfully trained on short temporal sequences (Nguyen-Duc-Thanh et al. (2012)). In our case, 10 closure levels and 4 speed levels were selected to cover the relevant operating space without unnecessary redundancy. Given the limited model order (maximum 4 hidden states), the parameter set remains tractable and does not require very large datasets.

Thereafter, we tested the shared-autonomy architecture with three possible numbers of hidden states (i.e., 2, 3, and 4 states). To select the best parameter set $\Theta^* = (A^*, B^*, \Pi^*)$ for each model, an ad-hoc nested cross-validation (nCV) was carried out. To better organize the nCV, the tactile dataset was divided into four distinct sets, each consisting of 10 grasps with 10 different closure levels performed at the same speed, as illustrated in Fig. 5.5. Therefore, the nCV consisted of four outer loops, each considering a different velocity level as the outer test set. For each iteration, the outer training set was further split into three folds, and a 3-fold

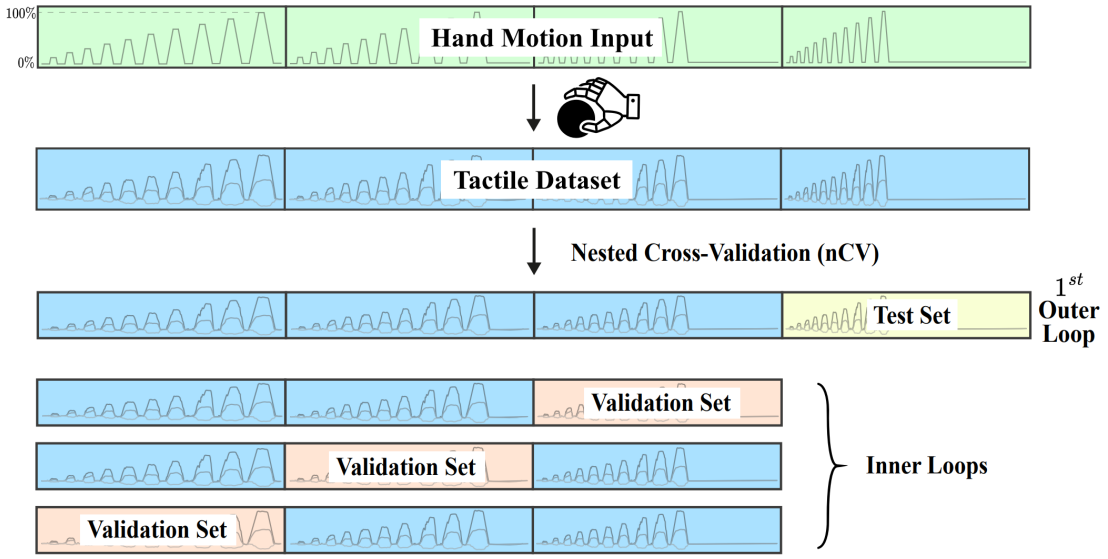


Figure 5.5 Schematic diagram of tactile dataset generation and HMMs training. Hand motion inputs were obtained by combining ten closure levels (from 0% to 100% with 10% increments) with four joint velocities, resulting in 40 grasp executions per robotic hand. Each trial generated a continuous temporal sequence of tactile measurements, later used to train and evaluate the HMMs through a 4-fold nCV.

CV was performed to select model hyperparameters. Finally, for each outer partition, the corresponding test set was used to evaluate the model winning the inner CV, resulting in four model evaluations for each nCV.

Experiment with Fixed Hand-Object Configuration

The goal of the first experiment was to systematically evaluate the performance of the proposed architecture in terms of grip-strength regulation. Four models were compared: the shared-autonomy architecture with 2, 3, and 4 hidden states, and the constant-gain architecture, to determine the best-performing configuration. In daily-life grasping activities, the ability to modulate grip strength is essential to avoid damaging fragile objects. Accordingly, participants were instructed to control the robotic hand via myoelectric signals, grasp an object (a plastic ball of diameter 6 cm for the AR10 and a plastic cylinder of diameter 3 cm for the industrial gripper), and reach a desired strength. Three target levels were selected ($\bar{f}_{\text{ext}}$ equal to 1.0, 1.5, and 2.0 N for the AR10 hand and $\bar{f}_{\text{ext}}$ equal to 1.0, 1.5, and 2.0 V for the industrial gripper) to represent increasing grip demands within the tested hardware range. To define these values, we equally divided the empirically observed force range so as to obtain appreciable force modulations. To account for sensor noise, a tolerance zone

$\bar{f}_{\text{ext}} \pm 0.1$ was introduced, and the two vibrotactile motors encoded the distance from this zone, one for values above and one for values below the target. For both motors, larger distances produced a higher duty cycle. To reduce inter-subject variability, participants were instructed to execute smooth grasps and progressively increase hand closure. Task success was confirmed if the user remained within the tolerance zone for 5 s before moving to the next level. Results are reported in Section 5.5.3.

Pick-and-Place Experiment

In the previous experiment, all models were compared in a controlled setup, where the metrics defined in Section 5.5.1 were minimally affected by contingent factors such as hand-object relative pose or incidental object motion. The present experiment therefore compared the best-performing variable-gain model from the fixed setup against the constant-gain architecture in a more realistic scenario, with the aim of assessing transfer to different objects and dynamic manipulation. Participants performed pick-and-place tasks with 9 objects of varying sizes and weights (3 cubes, 3 cylinders, and 3 spheres, each in three different dimensions) without exceeding a desired strength value. The task was intended to approximate the practical situation of grasping and moving a fragile object without crushing it. In this scenario, the control component of the architecture is particularly relevant because avoiding force overshoot is difficult without suitable modulation, especially when the feedback modality is kept identical across the compared controllers. For the AR10, the hand was mounted on a wearable structure that allowed the user to move it while grasping (Fig. 5.4, panel (a)). Participants performed the following sequence: (i) reach the object, (ii) grasp it with a tripodal grasp, (iii) lift and move it above a 10 cm-high cardboard wall, (iv) place it carefully on the other side of the wall, and (v) return to the starting position. For the Robotiq gripper, which was secured as end-effector of the Panda robot, the user controlled the gripper via myoelectric signals while the robot relocated the grasped item (Fig. 5.4, panel (b)). To enforce robustness, three different strength levels ($\bar{f}_{\text{ext}}$ equal to 1.0, 1.5, and 2.0 N) were selected for each object. Any repetition in which the user exceeded the target value was marked as failed. The same feedback modality used in the previous experiment was retained, using a single vibration motor because forces above the threshold were treated only as failures. Although exact target-force knowledge is seldom available in practice, prescribing common reference levels made the comparison between control strategies more interpretable. Results are reported in Section 5.5.6.

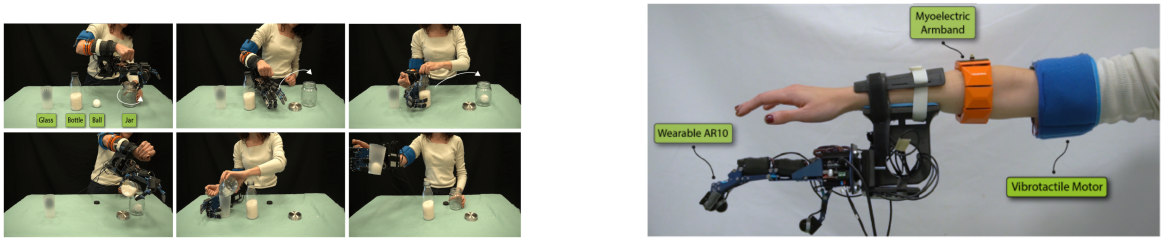
Recipe Preparation Experiment

As real-world applications are more complex and include stronger dynamic effects than the previous setups, the last experiment drew inspiration from an everyday recipe-preparation activity to further validate the approach in a more realistic scenario. Participants were instructed to sequentially grasp a series of objects with different target strength levels, denoted as LOW, MEDIUM, and HIGH, while performing the following sequence of operations with the wearable AR10 hand equipped with Optoforce sensors:

- Grasp the slippery lid of a jar with HIGH strength, then unscrew it and open the container.
- Gently (LOW strength) grasp a small, lightweight ball and place it inside the jar.
- Grasp a deformable bottle with MEDIUM strength to unscrew the cap.
- Increase the strength applied to the bottle to the HIGH level and then pour the contents into the jar.
- Grasp a plastic glass with LOW strength and pour the contents of the jar into it.
- Increase the strength applied to the plastic glass to MEDIUM to deliver the glass to an external user.

No explicit numerical value was assigned to each level. Instead, participants interpreted these levels through vibrotactile feedback, whose duty cycle was proportional to the estimated grip strength. Users were therefore instructed to steadily grasp each object at the level they judged appropriate and then continue the task. Repetitions were marked as failed if the object was not held firmly enough, if a re-grasp was required, or if excessive force visibly crumpled the deformable bottle. As in the pick-and-place study, the 3-state shared-autonomy architecture, which had shown the best performance in the fixed experiment, was compared against the constant-gain baseline. Since the earlier experiments had already quantified overshoot reduction and object transfer, the main purpose of this task was to assess whether the framework still enabled distinguishable strength regulation in a more complex sequence of actions. To obtain preliminary evidence on transfer to users with limb loss, the same protocol was replicated with one participant with transradial limb loss. In this case, the setup was adapted to the available clinical constraints. Because the AR10 is not a certified medical device, the Hannes hand, a CE-marked anthropomorphic hand with one opening/closing degree of freedom, was used and secured on a support. Grip strength was estimated from

a single Optoforce sensor placed on the thumb, which provides a practical approximation under the selected hand configuration. The participant executed the same object sequence, with the objects sequentially presented close to the device, for a total of 10 repetitions. Given the single-participant nature of this condition, the resulting evidence is interpreted as pilot only. Results are reported in Section 5.5.7.



(a) (b)

Figure 5.6 Setup for the Recipe Preparation Experiment with (a) the AR10 robotic hand. The user wears a myoelectric armband to control the robotic device with the presented shared-autonomy strategy and receives vibrotactile feedback on the same arm (b). Finger tactile sensors provide real-time measurements used for grip strength estimation and phase prediction. To complete the task, the user was instructed to sequentially grasp a series of objects with different strength levels –denoted as LOW, MEDIUM, and HIGH– to perform a recipe preparation task.

5.5 Performance Analysis and Stability Considerations

5.5.1 Metrics

Performance in the fixed hand-object experiment was assessed through completion time, grip-strength overshoot, and muscular effort. Specifically: completion time $T_c = T_{\text{end}} - T_{\text{init}}$, where $T_{\text{init}}, T_{\text{end}} \in \mathbb{R}$ are start and success times; overshoot $f_{\text{os}} = \max\left(f_{\text{ext}}(t) - \bar{f}_{\text{ext}}\right)$, where f_{ext} is estimated grip strength and $\bar{f}_{\text{ext}}$ is target force; muscular effort $M_e = \int_{T_{\text{init}}}^{T_{\text{end}}} |\sigma(t)|\, dt$, with $\sigma(t)$ the user velocity intent. This effort metric is used as an operational proxy for cumulative user engagement, because the modulated intent captures both the magnitude and duration of the control action generated from the sEMG interface. In our implementation, we integrated user velocity intent instead of average sEMG directly, because the NMF-based decomposition yields instant flexor/extensor activity and the resulting control input captures both intensity and duration of muscular engagement. With normalized intent (Eq. (10)), inter-user comparability is preserved. For the pick-and-place study, performance was summarized

by average success $AS \in \mathbb{R}$, defined as the average number of correctly completed pick-and-place actions (no visible slip and no target overshoot), in line with practical manipulation criteria.

5.5.2 Offline Evaluation of HMMs

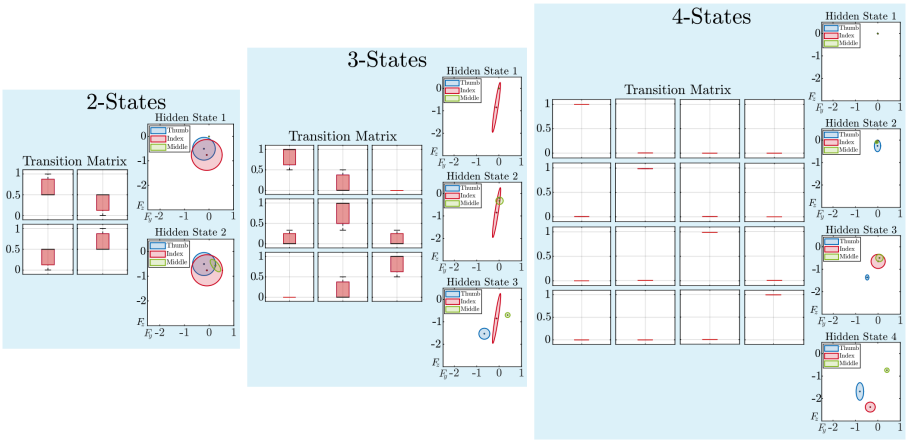
The results of the nCV, reflecting HMM generalization performance for both the AR10 and the gripper (considering 2, 3, and 4 hidden states), are summarized in Table 5.1. The evaluation metric was the number of correctly predicted grasping sequences in the nCV test set. Since the goal of the architecture is to modulate shared-autonomy based on the inferred grasp phase, a correct probabilistic description of the sequence is essential for smooth assistance. Given the sequential nature of the grasping process, we considered a hidden-state prediction correct when it described the grasp as a progression through adjacent HMM states, each associated with a distinct interaction phase. Table 5.1 therefore reports the mean number, and standard deviation, of correctly identified grasping actions, averaged across the 4 outer loops of the nCV. Each outer test set consisted of 10 grasps, which is therefore the maximum attainable score per split.

Table 5.1 HMMs Generalization Performances: Average Number (and Standard Deviation) of Correctly Identified Grasping Actions.

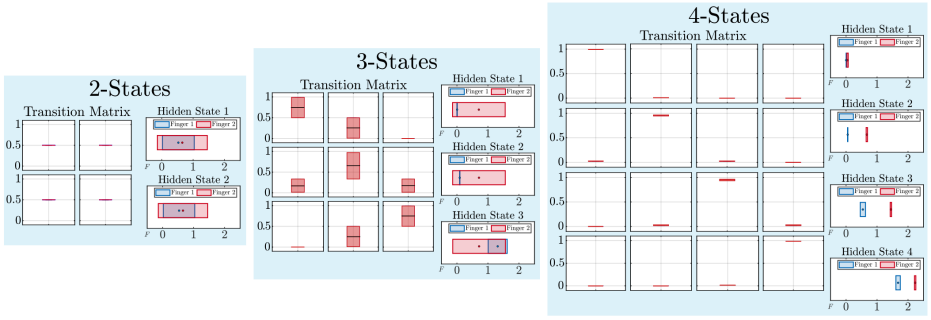
	2 States	3 States	4 States
AR10	8.67 ± 1.54	9.33 ± 1.02	5.50 ± 1.37
Gripper	8.75 ± 3.06	8.75 ± 2.32	9.00 ± 1.12

A non-parametric Friedman test was conducted to investigate the effect of Model Type (2 states, 3 states, and 4 states) on HMM performance. For the AR10 hand, the test revealed a statistically significant influence of Model Type on the number of correctly identified grasps ($\chi^2(2) = 11.526$, $p = 0.003$). A post-hoc Wilcoxon signed-rank test with Bonferroni correction (adjusted significance threshold $\alpha = 0.05/3 = 0.0167$) then showed that the 2-state and 3-state models outperformed the 4-state model, suggesting that increasing the number of states beyond three did not improve grasp-sequence identification on this dataset. In contrast, the Friedman test conducted on gripper results did not show a statistically significant difference between the three models. A plausible explanation is the structural simplicity and repeatability of the industrial gripper, whose lower variability may already be captured by lower-order models. To support this interpretation, the optimal HMM parameters obtained through the nCV procedure $\Theta^* = (A^*, B^*, \Pi^*)$ are visualized in Fig. 5.7, panel

(a), for the AR10 and panel (b) for the parallel gripper. The figure reports the learned transition matrices A^* and Gaussian observation parameters $B^* = \{\mu^*, \Sigma^*\}$, providing a qualitative view of how the models encode the temporal evolution of tactile data. For each model configuration, the transition matrices of the best-performing model over the 4 outer loops are aggregated for the different fingers and represented as boxplots. Similarly, the Gaussian emission parameters are visualized for each state in terms of estimated mean μ^* and covariance Σ^*, separately for each finger; for visualization purposes, only the features F_y and F_z are shown for the AR10 hand.



(a)



(b)

Figure 5.7 Qualitative visualization of the optimal HMM parameters obtained through 4-fold nCV for the (a) AR10 hand and the (b) parallel gripper. For each model configuration (2, 3, and 4 states), the transition probabilities of the best-performing model over the 4 outer loops are aggregated for the different fingers and represented as boxplots. The Gaussian emissions parameters are visualized in terms of estimated mean μ^* and covariance Σ^*, separately for each finger.

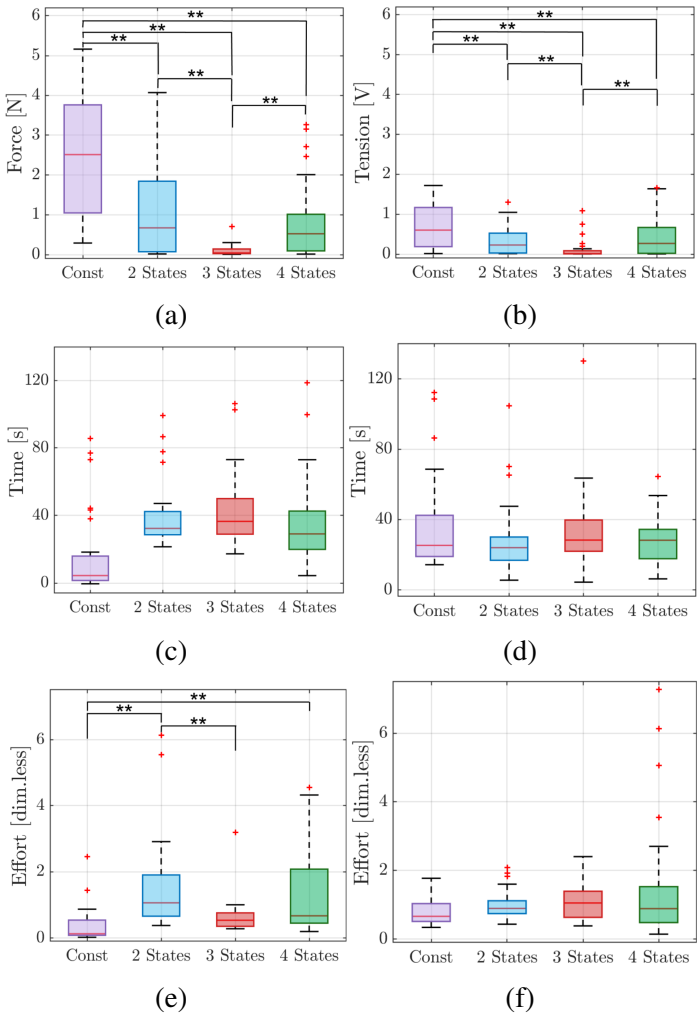


Figure 5.8 Boxplots of aggregated results for the Experiment with Fixed Hand-Object Configuration, for the AR10 hand (a)-(c)-(e) and the industrial gripper (b)-(d)-(f). First row: strength overshoot f_{os}. Second row: completion time T_c. Third row: muscular effort M_e. Symbol "**" indicates a statistically significant difference.

5.5.3 Experiment with Fixed Hand-Object Configuration

Statistical analysis was conducted on the results of the fixed hand-object experiment in terms of grip-strength overshoot, completion time, and muscular effort. The outcomes are shown in Fig. 5.8. For each metric, a nonparametric Friedman test was conducted for the factor Model Type (constant gain, 2 states, 3 states, and 4 states), since the Shapiro–Wilk test indicated non-normality in the considered groups. The test on overshoot revealed a statistically significant influence of Model Type for both the AR10 ($\chi^2(3) = 41.280$, $p < 0.001$) and the industrial gripper ($\chi^2(3) = 34.920$, $p < 0.001$); see Fig. 5.8, panels (a) and (b). A Wilcoxon signed-

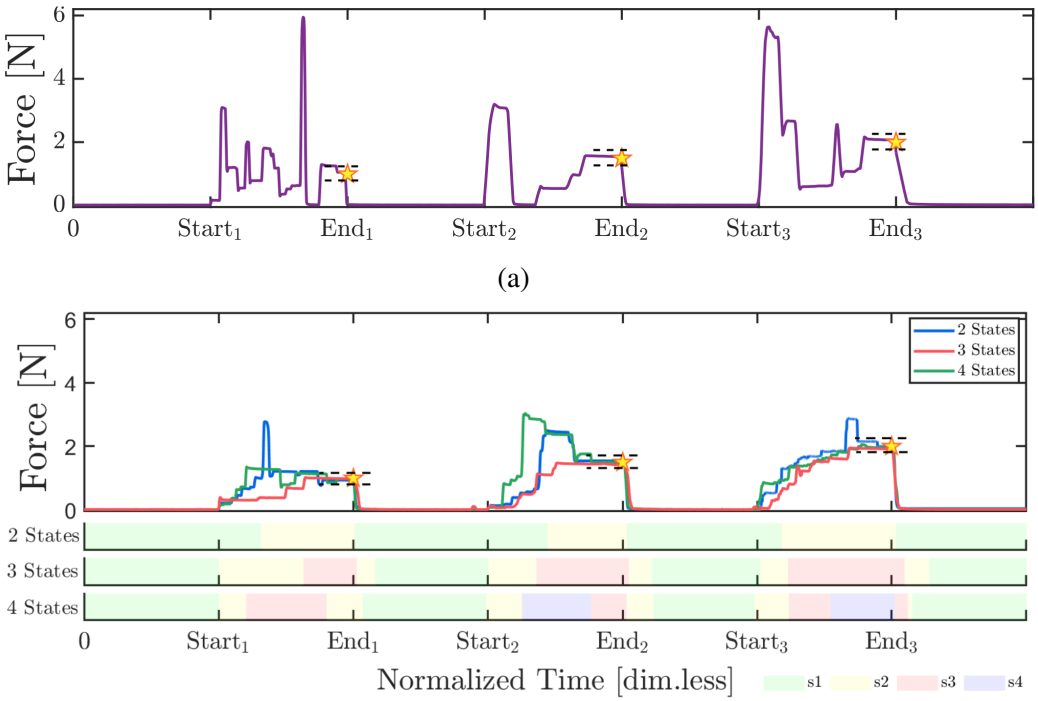


Figure 5.9 Experiment with Fixed Hand-Object Configuration for the AR10 hand. (a) Time evolution of the estimated grip strength $f_{\text{ext}}(t)$ for the constant gain architecture. (b) Top: time evolution of the estimated grip strength $f_{\text{ext}}(t)$ for the variable gain architectures with 2, 3, and 4 hidden states. Bottom: hidden-state evolution diagram for the three variable-gain architectures. Success is marked with a star.

rank test with Bonferroni correction (adjusted significance threshold $\alpha = 0.05/6 = 0.0083$) showed that the 3-state shared-autonomy architecture performed better, in terms of overshoot avoidance, than the constant-gain baseline and the shared-autonomy architectures with 2 and 4 states. This behavior is consistent with a 3-state interpretation in which state 1 captures free-space approach, state 2 initial contact establishment, and state 3 fine force modulation after contact. Under this interpretation, the 4-state model appears to over-segment the interaction phases, increasing switching and overshoot sensitivity on the available dataset. The same analysis conducted on completion time indicated that Model Type did not significantly affect the time required to complete the task for either robotic device; see Fig. 5.8, panels (c) and (d). This suggests that the gain reduction used to improve contact sensitivity did not produce a measurable penalty in completion time under the tested conditions. Muscular-effort results showed different trends for the AR10 and the industrial gripper. No significant effect of Model Type was found on gripper results (Fig. 5.8, panel (f)), whereas the test conducted on AR10 results revealed statistically significant differences in effort means

($\chi^2(3) = 43.400$, $p < 0.001$); see Fig. 5.8, panel (e). A further post-hoc comparison with Bonferroni correction (adjusted significance threshold $\alpha = 0.05/6 = 0.0083$) showed that both the constant-gain and the 3-state shared-autonomy architectures significantly reduced the muscular effort required to complete the task with respect to the 2-state architecture, with no statistically significant difference between these two configurations. The constant-gain architecture also outperformed the 4-state architecture, while no statistically significant difference was observed between the 3-state and 4-state architectures. To further interpret these results, Fig. 5.9 reports the time evolution of the estimated grip strength during the experiment conducted by Subject 10 with the AR10 hand. For each sequence of three levels, the time axes were stretched to superimpose the start and end times of each level (Start_i, End_i with $i = 1,2,3$), providing an intuitive visualization of state evolution across models. The overshoot behavior of the constant-gain baseline (Fig. 5.9, panel (a)), characterized by abrupt fluctuations of the estimated grip strength, is visibly attenuated with the shared-autonomy architectures (Fig. 5.9, panel (b)). The lower part of Fig. 5.9, panel (b), further shows the hidden-state evolution for the three shared-autonomy configurations. In particular, delayed identification of state two in the 2-state model and less smooth transitions in the 4-state model offer a plausible explanation for their weaker overshoot performance.

5.5.4 Gain Effect Analysis

To provide a qualitative analysis of the effect of gain parameters on system performance, we repeated the static experiment with the AR10 hand while systematically varying the gain values. Two scenarios were investigated: varying the maximum gain $k_{\max}$ while keeping the minimum gain $k_{\min}$ constant, and varying $k_{\min}$ while keeping $k_{\max}$ constant. In the first scenario, $k_{\min}$ was fixed at 0.1 and $k_{\max}$ was varied between 1 and 3. In the second scenario, $k_{\max}$ was fixed at 2 and $k_{\min}$ was varied over 0.001, 0.01, 0.1, 1, and 2. As in the fixed hand-object experiment, the robotic hand was commanded to reach the three target force levels ($\bar{f}_{\text{ext}}$ equal to 1.0, 1.5, and 2.0 N) and remain within the tolerance zone for 5 s. To improve repeatability, human control was replaced with a constant speed reference ($\sigma = 0.1$), so that the comparison was not affected by operator variability. Results are reported in Fig. 5.10. Panels (a) and (b) show the effect of varying $k_{\max}$ on completion time and strength overshoot, respectively, while panels (c) and (d) show the corresponding effect of varying $k_{\min}$. Increasing $k_{\max}$ reduced completion time but increased overshoot. Conversely, reducing $k_{\max}$ reduced overshoot at the cost of slower execution. Lowering $k_{\min}$ also reduced overshoot, but tended to increase completion time because the system moved

more slowly once contact had been established. No statistical analyses were performed on these results, as their purpose was only to provide a qualitative view of how $k_{\max}$ and $k_{\min}$ shape system behavior. The boxplots support the intended trade-off: higher gains improve responsiveness, whereas lower gains improve contact sensitivity.

5.5.5 Grasp Quality

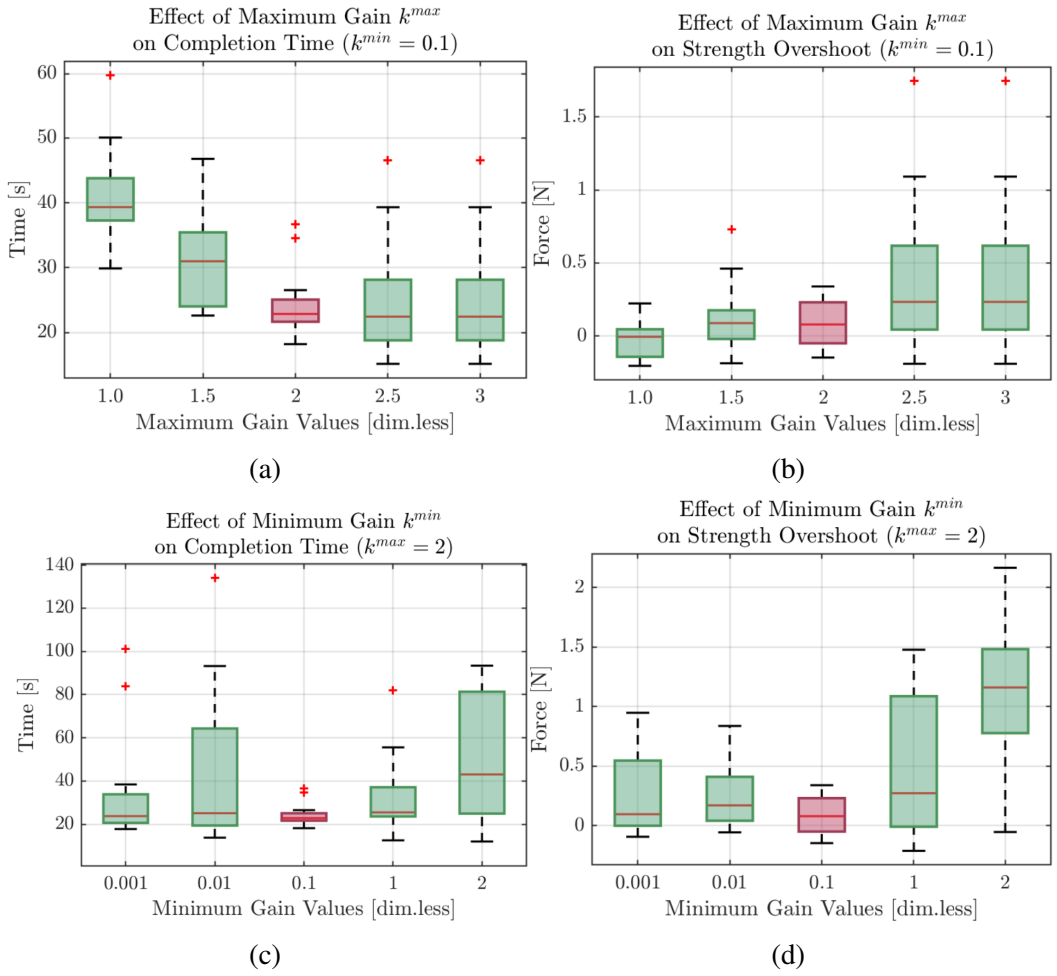


Figure 5.10 Gain Effect Analysis: impact of varying $k_{\max}$ (first row) and $k_{\min}$ (second row) on system performance, expressed in terms of completion time (a), (c) and strength overshoot (b), (d). The red boxes represent the gains used in our architecture.

To assess the effectiveness of the proposed strategy in terms of grasp quality, we defined the risk-of-instability index as $RI(t) = \|f_T(t)\|/\|f_N(t)\| \in \mathbb{R}$, namely the ratio between the norm of tangential forces $\|f_T(t)\|$ and normal forces $\|f_N(t)\|$ measured by the Optoforce sensors. This ratio has previously been used in (Kaboli et al. (2016)) to detect slippage of

the grasped object. Here, it is adopted as a quantitative proxy for grasp quality because it estimates how close the resultant contact force is to the friction-cone boundary. Lower values therefore indicate more stable grasps. We analyzed the fixed hand-object results using this metric. For an initial qualitative interpretation, the risk-of-instability index was computed along the full three-level grasp sequence for a representative subject. As shown in Fig. 5.11, panel (a), the index for the 3-state shared-autonomy architecture decreases after contact establishment and settles once the estimated grip strength reaches a stable value. Moreover, the settled value decreases as the target grip-strength level increases, which is consistent with larger normal forces moving the resultant force farther from the friction-cone limit. In contrast, the constant-gain architecture exhibits larger oscillations in the index (Fig. 5.11, panel (b)), which may reduce grasp robustness. Statistical analysis supported this trend. A nonparametric test conducted on the aggregated results, using the final value of the risk-of-instability index for each grip-strength level over the 10 subjects, showed a statistically significant difference across levels: ($\chi^2(2) = 14.600$, $p < 0.001$) for the thumb, ($\chi^2(2) = 18.200$, $p < 0.001$) for the index, and ($\chi^2(2) = 18.200$, $p < 0.001$) for the middle finger. A further post-hoc comparison with Bonferroni correction (adjusted significance threshold $\alpha = 0.05/3 = 0.0167$) indicated that all levels (L1, L2, L3) differed from one another; see Fig. 5.12. Finally, to compare the shared-autonomy framework against the constant-gain baseline in terms of excursions toward potentially unstable grasp conditions, we computed the maximum value of the risk-of-instability index $RI_{\max}$ for each grasp repetition, starting from the instant at which the estimated grip strength first crossed the lower bound of the target reference. Statistical analysis revealed a significantly lower maximum risk of instability for the 3-state architecture ($\chi^2(1) = 7.12$, $p < 0.01$); see Fig. 5.13.

5.5.6 Pick-and-Place Experiment

Figure 5.14 reports the average success *AS* in the pick-and-place experiment, defined in Section 5.5.1 as the average number of successful pick-and-place operations over three repetitions, across the 10 participants and for both constant-gain and 3-state shared-autonomy control. A Wilcoxon signed-rank test was used to evaluate the effect of Model Type (constant gain vs. 3-state), since normality could not be assumed. Results showed higher average success for the 3-state shared-autonomy controller across the tested objects. At the same time, both controllers showed object-level transfer within the considered set, with no significant differences among object categories, indicating that shape and weight did not markedly bias performance under either strategy.

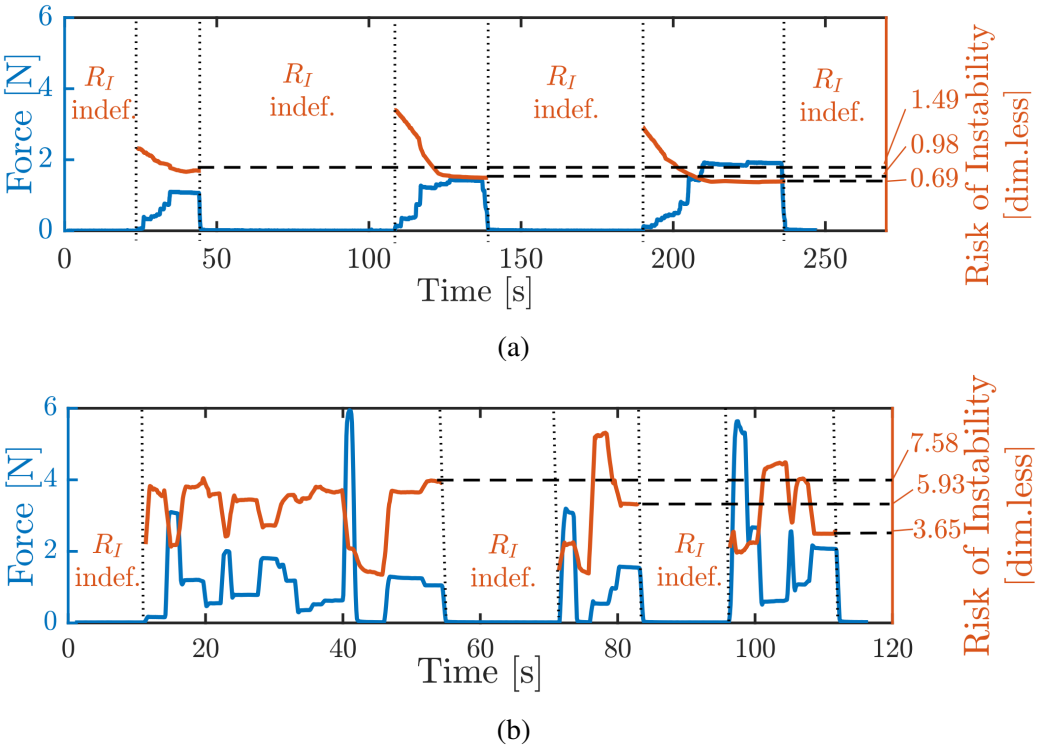


Figure 5.11 Grasp Quality: time evolution of the estimated grip strength $f_{\text{ext}}(t)$ (blue) and of the thumb risk-of-instability index $RI(t)$ (orange) for a sequence of three-level grasps for (a) the 3-state shared-autonomy architecture and (b) the constant gain architecture.

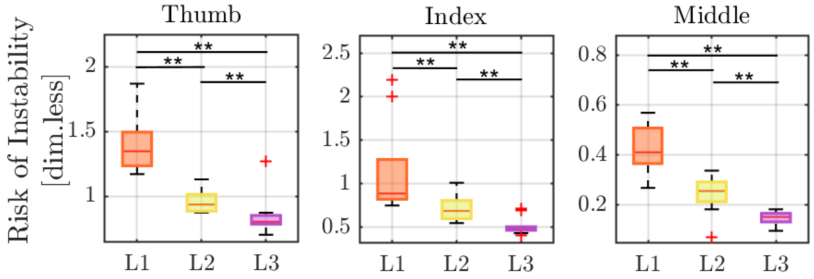


Figure 5.12 Grasp Quality: boxplots of risk-of-instability index *RI* aggregated results for thumb, index, and middle fingers, for the three levels (denoted as L1, L2, L3).

5.5.7 Recipe Preparation Experiment

Since the previous experiments had already assessed grip-strength overshoot, completion time, muscular effort, and transfer across object geometries, the recipe-preparation experiment focused on the ability to regulate different grip-strength levels within a more complex sequence of actions. The first row of Fig. 5.15 reports the aggregated grip-strength values obtained by able-bodied participants across the three target levels (LOW, MEDIUM, and

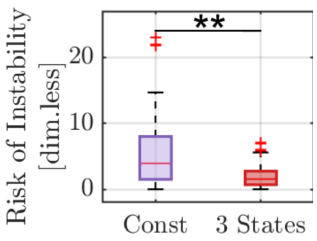


Figure 5.13 Grasp Quality: boxplots of aggregated maximum risk-of-instability index $RI_{\max}$ for the 3-state shared-autonomy and the constant gain architectures.

HIGH), for both the constant-gain architecture (panel (a)) and the 3-state shared-autonomy architecture (panel (b)). To enable comparisons, results were normalized with respect to each participant's maximum recorded value. Trials performed with the constant-gain architecture exhibited larger variability, especially at the MEDIUM level, whereas the 3-state shared-autonomy architecture produced more clearly separated distributions. A nonparametric Friedman test was conducted to investigate whether the produced grip forces differed across the three target levels, since the Shapiro–Wilk test indicated non-normality. The analysis revealed a statistically significant difference across strength levels for both the constant-gain architecture ($\chi^2(2) = 28.778$, $p < 0.001$) and the 3-state shared-autonomy architecture ($\chi^2(2) = 38.100$, $p < 0.001$). A post-hoc pairwise comparison with Bonferroni correction should therefore be interpreted with an adjusted significance threshold of $\alpha = 0.05/3 = 0.0167$. Under this criterion, the constant-gain architecture separated LOW from the other levels, whereas the distinction between MEDIUM and HIGH was weaker. By contrast, all grip-strength levels were separated for the 3-state shared-autonomy architecture, supporting the more cautious claim that this controller enabled finer level discrimination in the tested task. Results from the participant with limb loss are shown in the second row of Fig. 5.15. For the 3-state shared-autonomy architecture, the Friedman test revealed a statistically significant difference across strength levels ($\chi^2(2) = 20.00$, $p < 0.001$). In contrast, the Friedman test conducted on the constant-gain results was not significant ($\chi^2(2) = 5.60$, $p < 0.061$). Because this condition involved a single participant and an adapted setup, these observations should be regarded as pilot evidence rather than as a generalizable comparison. Failure analysis showed more failures for the constant-gain architecture, especially when grasping the slippery jar lid and the deformable plastic bottle. For able-bodied participants, finger slippage while grasping the jar lid occurred 3 times with the shared-autonomy architecture and 5 times with the constant-gain one; plastic-bottle crumpling or slippage occurred 3 times with shared-autonomy and 7 times with constant gain; glass re-grasp due to excessive force

occurred 3 times for both architectures. In the pilot experiment with the participant with limb loss, finger slippage occurred 3 times for both architectures, whereas plastic-bottle crumpling or slippage occurred 6 times with shared-autonomy and 11 times with constant gain. No glass failures occurred for either architecture. Within the limits of this experiment, these observations are consistent with a finer regulation of interaction forces under the shared-autonomy controller, including with deformable or slippery objects.

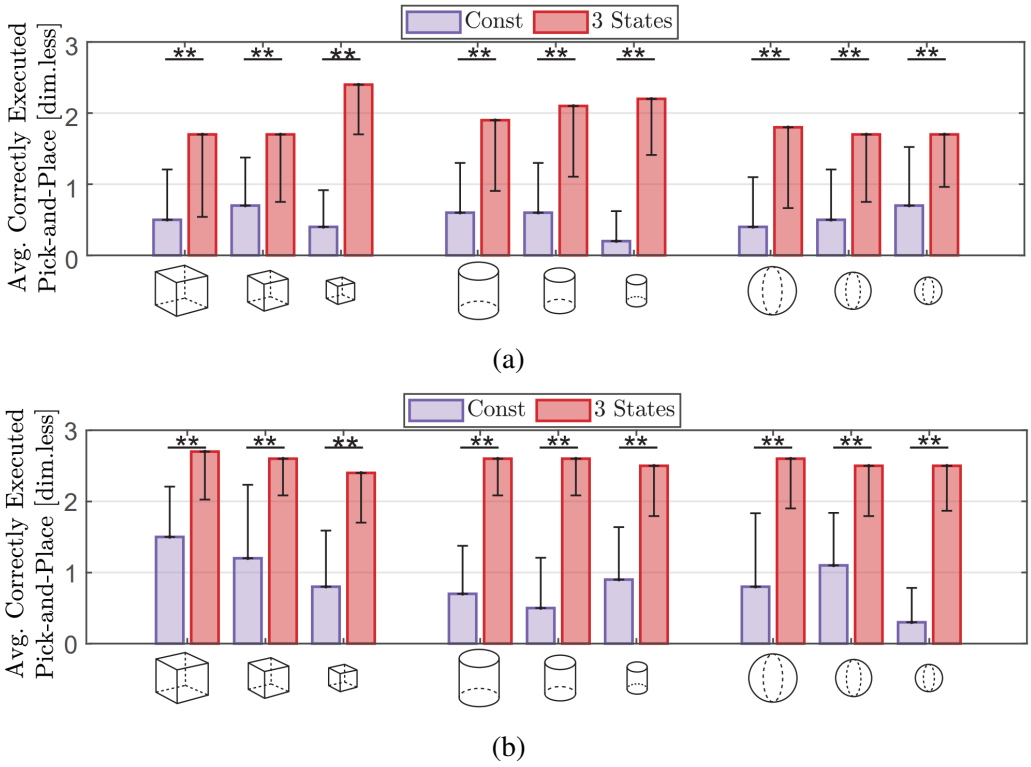


Figure 5.14 Pick-and-Place Experiment for the AR10 hand (a) and the industrial gripper (b). Average success *AS* across 10 participants for the constant gain (violet) and the 3-state shared-autonomy architecture (red). Symbol "**" indicates a statistically significant difference.

5.5.8 Subjective Evaluation of the Architecture

To qualitatively evaluate the proposed shared-autonomy architecture, we asked users to fill out a questionnaire in which some statements about the tested framework had to be rated on the basis of a Likert scale from 1 (entirely disagree) to 7 (entirely agree). Specifically, the questionnaire included four statements related to the perceived ease of use (PE), perceived usefulness (PU), emotion (E), and comfort (C) in regulating the grip strength with the proposed framework.

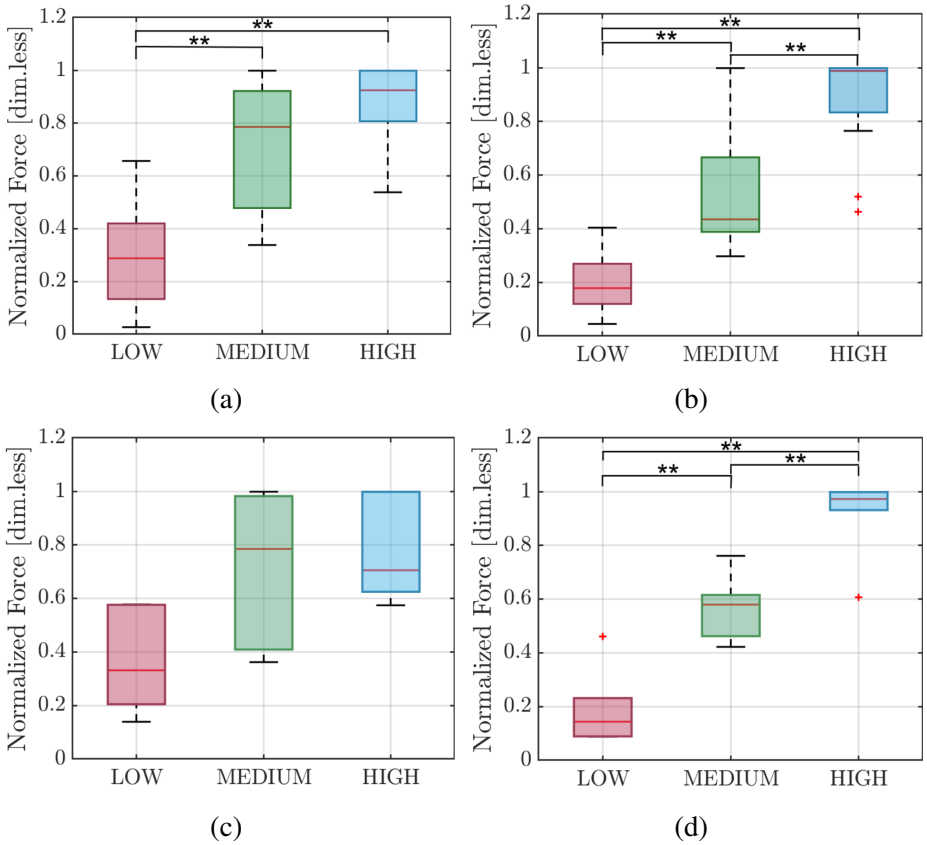


Figure 5.15 Recipe Preparation Experiment. First row: boxplot of normalized grip-strength values (LOW, MEDIUM, and HIGH) for the constant gain (a) and the 3-state shared-autonomy architecture (b) for the experiments performed by able-bodied participants. Second row: boxplot of normalized grip-strength values for the experiments performed by the amputee subject (constant gain architecture: (c), shared-autonomy architecture: (d)). Symbol "**" indicates a statistically significant difference.

As reported in Table 5.2, all average scores except the one concerning vibrotactile feedback were greater than or equal to 6, indicating an overall positive qualitative evaluation of the proposed shared-autonomy architecture for grip-strength regulation. Users were less positive about the ease of interpreting the vibrotactile feedback. In particular, the statement regarding the appropriateness of vibrotactile feedback received a relatively low score (5.8 ± 0.42 for the pick-and-place experiment and 5.8 ± 0.63 for the recipe task), indicating lower confidence in feedback interpretation. This may be due to subjective variability in sensitivity to vibrotactile stimuli, the cognitive effort required to associate vibration with grip-strength modulation, or mechanical factors such as motor placement and arm-contact quality.

5.6 Limitations and Future Extensions

Table 5.2 Qualitative Evaluation of the Architecture.

Outcome Type	**Statement**	**Average Score ± Std. Dev. Pick-and-Place**	**Average Score ± Std. Dev. Recipe Task**
PE	*It was easy to modulate the grip strength with the shared-autonomy approach.*	6.5 ± 0.53	6.1 ± 0.57
PU	*I think the shared-autonomy approach allows to regulate the grip strength to different target levels.*	6.7 ± 0.48	6.3 ± 0.47
E	*I liked to modulate the grip strength using the shared-autonomy approach.*	6.3 ± 0.28	6.0 ± 0.5
C	*It was appropriate to have the vibrotactile feedback on the upper arm.*	5.8 ± 0.42	5.8 ± 0.63

A well-known limitation of sEMG-based control solutions is their sensitivity to various external factors, such as electrode shift, fatigue, and motion artifacts. These aspects have not been systematically analyzed in this chapter, whose primary objective is to validate the feasibility of the presented shared-autonomy control architecture, rather than its long-term usability. Future works will address extended usage scenarios and associated challenges, with the objective of evaluating the robustness and stability of the system over time. Another limitation of the approach is the relatively limited sample size adopted in the experiments. To investigate the statistical significance of the reported results, a sensitivity analysis was performed on the within-subjects design using the software G*Power to determine the smallest effect size detectable given the sample size ($N = 10$), α ($\alpha = 0.05$), and power level ($1 - \beta = 0.8$). The analysis showed a minimum detectable effect of $f = 0.397$ for overall comparisons and $d_z = 0.996$ for pairwise comparisons, indicating that the design is sensitive to moderate-to-large effects. On the other hand, small-to-medium effects would likely remain undetected with such a limited sample size. To detect medium effects (effect size $d_z = 0.5$ based on Cohen's rule), while maintaining the same α and power level values, the required sample size would have been approximately $N = 34$ subjects. A key limitation of the proposed solution is its dependence on the selected control gains. Indeed, the performance of the overall framework is highly sensitive to the chosen gain values. To address this aspect, two critical working requirements essential for human-driven manipulation tasks were identified: reactivity during non-contact phases and sensitivity during in-contact phases. These requirements were used to develop a systematic and repeatable approach to gain selection, in order to obtain consistent performances across tasks. Furthermore, an analysis of how gains affect system performance is presented, supporting the proposed approach. However, future works could address the aforementioned limitation by exploring adaptive gain-tuning strategies or learning-based approaches to dynamically adjust control parameters in real-time, further enhancing the robustness and versatility of the system. Another future

research direction involves the integration of user intention and joint velocity information into the HMM formulation. This enhancement aims to improve prediction capabilities and make the user an active part of the control. Other sensing equipment and robotic hands will be explored to test the framework and offer more evidence of its robustness with respect to different hardware solutions. Finally, since a small population of 10+10 able-bodied subjects and 1 participant with upper limb loss was considered in this study, future studies will involve a larger number of users to create a broader dataset for analysis.

5.7 Concluding Methodological Remarks

This chapter introduced a shared-autonomy framework for grip-strength regulation in HITL robot-hand control, addressing limitations of state-of-the-art approaches that often rely on lengthy supervised training and deterministic grasp-control assumptions. To handle real-world uncertainty, HMMs were used to model probabilistic tactile dynamics during grasping. Online phase probabilities were then mapped to adaptive shared-autonomy gains, modulating control authority between human operator and robotic hand. Three experiments were conducted: a systematic comparison among architectures with different numbers of hidden states, an object-generalization pick-and-place study, and a realistic recipe-inspired manipulation task. Across these evaluations, the proposed approach improved force-regulation behavior and provided a practical compromise between responsiveness and contact sensitivity, including in scenarios with deformable and slippery objects and in a pilot validation with a participant with limb loss.

Chapter 6

Hierarchical Probabilistic Skill Encoding

6.1 From Regulation to Skill Encoding

Chapter 5 showed that probabilistic phase inference can improve grip-force regulation by combining sEMG intent, tactile sensing, and adaptive gain modulation during contact-rich manipulation. This chapter builds on that mechanism but addresses a different uncertainty source and a broader objective. The uncertainty is no longer limited to online force regulation around a fixed task: it also concerns how a manipulation skill should be encoded, reproduced, and transferred once force becomes part of the demonstration. Accordingly, the chapter extends the regulation-oriented formulation of Chapter 5 toward hierarchical probabilistic control, force-aware programming by demonstration, and autonomous reproduction of learned skills. Its main limitation is that validation remains focused on a restricted set of grasping tasks and object variations.

Fine grip-force regulation remains difficult in robotic hands, especially in human-in-the-loop (HITL) settings where user intent, robot mechanics, and contact conditions interact during task execution. Robotic grasping research has made substantial progress, yet smooth and precise force regulation remains an open challenge (Mason (2018)). In HITL control, the objective is to transfer human adaptability and intuition to robot hands while maintaining natural interaction (Niemeyer et al. (2016)). Advanced human-robot interfaces support applications in teleoperation, rehabilitation, prosthetics, physical assistance, and collaborative manufacturing (Fleischer and Hommel (2008); Huang et al. (2021); Khatib et al. (2016)). Their effectiveness depends on usability and on how sensing and control algorithms support users in expressing grasping intentions (Laghi et al. (2020)).

Surface electromyography (sEMG) has emerged as a key sensing modality for extracting user intent (Meattini et al. (2018); Porges et al. (2019)). Pattern-recognition approaches based

on SVMs achieved good accuracy (Scheme and Englehart (2011)) but are limited to discrete gestures, preventing HITL continuous regulation (Fougner et al. (2012)). Simultaneous and proportional control (Farina et al. (2014); Gijsberts et al. (2014)) uses regression models for continuous multi-DOF control, enabling more natural adaptation. Unsupervised, semi-supervised, and self-supervised regression methods (Meattini et al. (2022, 2023a)) mitigate labeling demands. However, sEMG remains stochastic due to neuromuscular recruitment (Burdet et al. (2013)), causing small command fluctuations that become critical once contact is established because they may induce significant force variations (Prattichizzo et al. (2013)). In addition, non-ideal robot-hand mechanics, especially in complex anthropomorphic designs, further hinder precise HITL grip regulation. This chapter therefore focuses on extending force control into force-aware skill encoding under the same sensing constraints.

Tactile feedback is therefore important to complement visual cues and close the perception loop during grasping. Vibrotactile stimulation is widely studied (Antfolk et al. (2013); Childress (1980)) and can be integrated into wearable devices with limited hardware complexity (Cipriani et al. (2008); Pylatiuk et al. (2006)). In HITL robot hand control, tactile data has mainly supported slippage detection (Howe (1993); Romeo et al. (2021)), with more limited attention to continuous force regulation. Automatic force control has been explored for grasp stabilization and in-hand manipulation (Prattichizzo and Trinkle (2016)), but explicit consideration of contact forces within HITL interfaces remains underexplored. Ultimately, the success of HITL force control depends on how tactile forces correlate with user-derived commands, enabling controllable influence over inherently probabilistic grasp dynamics.

Deterministic grasp models struggle because they require accurate knowledge of contact geometry, object-hand relative motion, material properties, and interaction dynamics: parameters that are difficult to identify and often vary unpredictably during real manipulation. Probabilistic modeling is therefore well-suited to HITL grasping, especially because sEMG signals and, consequently, their machine-learning decoders are inherently stochastic, and human cognitive processing of artificial feedback is non-deterministic. These uncertainties yield poorly predictable relationships between input modulation and force output, contrasting users' expectations of linear and repeatable behavior. Embedding probabilistic modeling of the input–force relationship can thus improve controllability and user experience. In this context, shared autonomy provides a principled way to mediate the user's input through a probabilistic understanding of the grasping process. When the hand moves in free space, tactile information can indicate the absence of contact, allowing the system to remain fast and responsive. Once contact is detected, the probabilistic model can scale the input more

conservatively, enabling precise and stable grip-force regulation. In this way, the assistance level adapts smoothly to the user's commands, supporting both rapid hand motions and fine force adjustments when needed. Hidden Markov Models (HMMs) are well-suited for capturing temporal uncertainty and discrete interaction states, outperforming simpler methods like continuous regressors, which are more focused on continuous function estimation rather than the state transitions and temporal dynamics that are central in tasks requiring structured temporal dynamics. HMMs have been used to classify grasp sequences from tactile and glove data (Bernardin et al. (2003)) and to identify grasp strategies in cluttered environments using tactile and video information (Son (2022)). However, the problem of grip force regulation has not been explicitly addressed, and therefore remains largely unresolved.

Relative to regulation-only formulations, the objective of this chapter is broader: the same probabilistic mechanism used to modulate force online is extended to encode and reproduce manipulation skills. Thus, the scope moves from force regulation during execution to force-aware task programming. The elements inherited from Chapter 5 are the probabilistic interpretation of contact evolution and the adaptive modulation of user intent. The elements that are new here are: a hierarchical control organization, explicit force-aware skill encoding, a programming-by-demonstration formulation, and autonomous reproduction with limited object-size transfer.

In this chapter, an HMM-based probabilistic shared autonomy framework is presented that models the coupled relationship between sEMG-derived velocity commands and tactile measurements for fine human-in-the-loop grip force control. Velocity scaling is driven by inferred intent–force states, enabling rapid motion during non-contact phases and accurate force modulation upon contact. Validation on anthropomorphic and industrial robotic hands demonstrates improved stability, responsiveness, and adaptability, and supports programming by demonstration, with grip force encoded as a learnable parameter within an HMM-structured control hierarchy. This hierarchy is a key contribution: a lower layer models local intent–force coupling for real-time control, while a higher layer captures task-phase evolution for skill encoding and autonomous reproduction.

6.2 Hierarchical Probabilistic Control Architecture

Fig. 6.1 illustrates the overall HITL control architecture, comprising the human, robot, and probabilistic interaction-shaping subsystems. User intent from sEMG is mapped to robot velocity commands via a feedforward path, while tactile force information is returned through vibrotactile feedback. A central probabilistic shaping block modulates both signal paths

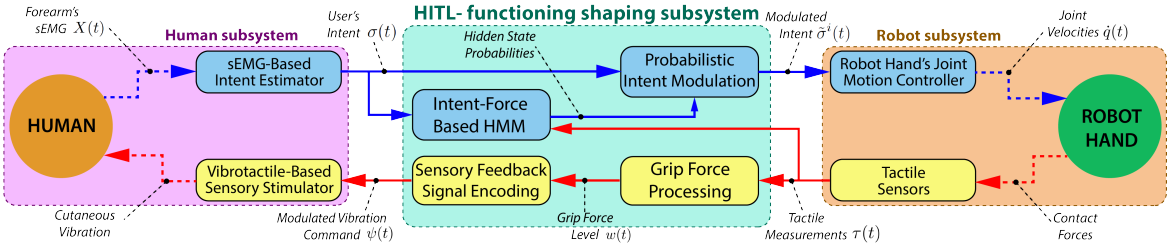


Figure 6.1 Overview of the HITL architecture showing intent estimation, feedback delivery, and HMM-based modulation for reliable grip force control.

to ensure consistent grip force regulation across grasping phases. Subsystem details are provided in the following.

6.2.1 Human Subsystem

The next two subsections describe the two blocks composing the human subsystem in Fig. 6.1.

sEMG-Based Intent Estimator

Assume an array of n sEMG sensors placed around the forearm (the *sEMG bracelet*), positioned near the main flexor and extensor muscle groups. The resulting n-dimensional signal is processed to estimate the hand-closure level during power grasps, yielding a rate-control velocity intent for regulating reaching, release, and grip force, as detailed later in this section. Therefore, in this regard, let us consider the vector of instantaneous Root Mean Square (RMS) sEMG values $X(t) \in \mathbb{R}^n$, obtained from raw sEMG signals by applying, to each channel, a processing sequence consisting of: a 50 Hz notch filter for powerline interference cancellation; a 20 Hz highpass filter to remove baseline noise; and the computation of the RMS using a 200 ms running window. During a grasping motion, the vector $X(t)$ can be expressed as the instantaneous linear mixture (Jiang et al. (2008))

$$X(t) = MA(t), \tag{6.1}$$

where $M \in \mathbb{R}^{n\times 2}$ is the *myoelectrical synergy matrix*, which maps the *myoelectrical activation vector* $A(t) = [a_o(t) \;\; a_c(t)]^T \in \mathbb{R}^{2\times 1}$, collecting the two predominant hand's opening and closing drives $a_o(t)$ and $a_c(t)$, respectively, into the RMS sEMG signals $X(t)$. Leveraging Eq. (6.1), it is possible to estimate the myoelectrical synergy matrix M, denoted as $\widehat{M}$, by applying the unsupervised factorization algorithm Non-negative Matrix Factorization (NMF) (Meattini et al. (2023a)) to a simple sEMG calibration recording $\overline{X} \in \mathbb{R}^{n\times d}$ (with d the number

of samples of the recording – note how $\overline{X}$ does not depend on time) acquired during the executing of a small set of grasping motion repetitions. In this case, five repetitions of the following motion sequence are considered: *(i)* start from the open hand posture; *(ii)* execution of grasp closing reaching the closed hand posture; *(iii)* execution of grasp opening returning to the open hand posture. Once $\widehat{M}$ is computed by applying NMF to $\overline{X}$, its pseudo-inverse $\widehat{M}^{+}$ can be used to compute an instantaneous estimation $\widehat{A}(t)$ of the myoelectrical activation vector from online sEMG signals

$$\widehat{A}(t) = \begin{bmatrix} \widehat{a}_o(t) \\ \widehat{a}_c(t) \end{bmatrix} = \widehat{M}^{+} X(t), \tag{6.2}$$

which can be used to compute the user's hand closure level $\zeta(t)$, obtained as the normalized difference between the estimated hand's opening and closing drives $\widehat{a}_o(t)$ and $\widehat{a}_c(t)$, respectively, as

$$\zeta(t) = \alpha(\widehat{a}_o(t) - \widehat{a}_c(t)) + \beta, \tag{6.3}$$

where α is a multiplicative scaling factor and β is an additive offset, both chosen to normalize $\zeta(t)$ into a suitable range according to control needs – in this case, such that $\zeta(t) \in [-1, 1]$. Let

$$\sigma_0(t) = \begin{cases} \zeta(t)\sigma_{\min}, & \zeta(t) < 0, \\ \zeta(t)\sigma_{\max}, & \zeta(t) \geq 0, \end{cases}$$

denote the pre-deadband intent signal. Then, in accordance with the desired rate control paradigm, the *grasping motion velocity intent* $\sigma(t)$ for robotic hand control is obtained as

$$\sigma(t) = \begin{cases} \sigma_0(t) - \sigma_{\text{db,inf}}, & \text{if } \sigma_0(t) \leq \sigma_{\text{db,inf}}, \\ 0, & \text{if } \sigma_{\text{db,inf}} < \sigma_0(t) < \sigma_{\text{db,sup}}, \\ \sigma_0(t) - \sigma_{\text{db,sup}}, & \text{if } \sigma_0(t) \geq \sigma_{\text{db,sup}}, \end{cases} \tag{6.4}$$

where $\sigma_{\text{db,inf}}$ and $\sigma_{\text{db,sup}}$ are proper bounds defining a deadband region to ensure that a static robot hand pose is held when the user rests their hand (i.e. in absence of muscle activity), avoiding the generation of unintentional commands due to small fluctuations in the sEMG signals, and, therefore, enabling a smooth and governable regulation of the robot hand's grasping motion velocity intent.

Vibrotactile-Based Sensory Stimulator

The HITL system incorporates a vibrotactile bracelet worn by the user on the upper arm, used to provide fundamental feedback about the grip force (see Sec. 6.2.3) applied by the robot hand to the grasped object. The vibrotactile bracelet consists of an armband with a vibration motor and related microcontroller that implements a Pulse Width Modulation (PWM) vibrational pattern, allowing for real-time feedback delivery through the user's skin stimulation. This enables informed adjustments of the control inputs (i.e. the grasping motion velocity command, see eq. (6.4)), based on the current level of grip force. Specifically, the robot hand's *grip force level* $w(t) \in \mathbb{R}$ is estimated based on tactile sensor readings and then mapped into a feedback signal $\psi(t)$ as detailed in the following in Sec. 6.2.3. In this way, the feedback signal $\psi(t)$ is then utilized as the modulation signal of a PWM-based carrier vibration pattern, ensuring a highly intuitive vibrational representation of the grip force level for the user.

6.2.2 Robot Subsystem

The following subsections introduce the two components forming the robot subsystem in Fig. 6.1.

Robot Hand's Joint Motion Controller

The robot hand grasping motion is controlled using a postural synergy based approach (Santello et al. (2016)). Specifically, considering a robot hand with r joints, the vector of joint velocities $\dot{q}(t) \in \mathbb{R}^r$ is computed as

$$\dot{q}(t) = \Sigma(t)P, \tag{6.5}$$

with $P \in \mathbb{R}^r$ the postural synergy (column) matrix (Santello et al. (2016)) and

$$\Sigma(t) = \text{diag}(\underbrace{\tilde{\sigma}^1, ..., \tilde{\sigma}^1}_{r_1 \text{ times}}, ..., \underbrace{\tilde{\sigma}^i, ..., \sigma^i}_{r_i \text{ times}}, ..., \underbrace{\tilde{\sigma}^m, ..., \tilde{\sigma}^m}_{r_f \text{ times}}) \tag{6.6}$$

the $r \times r$ matrix of velocity synergy activation values $\tilde{\sigma}^i = \tilde{\sigma}^i(t)$, for r_i finger's joints relative to the generic finger $i \in \{1, ..., f\}$. Since a single grasping motion is of interest in this case, the elements of P are simply imposed as the difference between the hand's fully extended and fully closed configurations q_{open} and q_{closed}, producing a coordinated closure/opening

motion of the robot hand's fingers to increase/decrease the grip force level. Regarding the input $\tilde{\sigma}_i(t)$ in Eq. (6.6), it represents the *modulated intent* obtained by shaping the grasping motion velocity intent $\sigma(t)$, previously introduced in Eq. (6.4), through the probabilistic interaction-shaping approach detailed in Sec. 6.2.3 and Eq. (6.9).

Tactile Sensors

The robotic hand is considered to be equipped with tactile sensors on the fingertips, with object grasping performed solely through contact forces applied by two or more of these fingertips. Therefore, in general, each tactile sensor provides a multidimensional tactile signal in response to the instantaneous contact force applied to the object. More precisely, let us consider $m \geq 2$ fingertips actively involved in the grasping process, each equipped with its own sensor. The *tactile measurement* is then defined $\tau(t) = [T_1(t)\ \ T_2(t)\ \ \dots\ \ T_m(t)]^T$ as the collection of each l-dimensional *tactile signal* $T_i(t) \in \mathbb{R}^l$ ($i = \{1, \dots, m\}$) associated to the single tactile sensors. As illustrated in Fig. 6.1, the tactile measurement $\tau(t)$ is then exploited in the HITL control scheme to both provide the user with a vibrational feedback encoding the grip force level information (see Sec. 6.2.1 for details), and to contribute to the shaping of the user's intent according to an HMM-based probabilistic modulation method, as reported in the subsequent section.

6.2.3 Probabilistic Interaction-Shaping Subsystem

The probabilistic interaction-shaping block (Fig. 6.1) processes the intent $\sigma(t)$ and tactile measurement $\tau(t)$ to produce the modulated command $\widetilde{\sigma}(t)$ and feedback $\psi(t)$ through an HMM-based adaptive mechanism explained in this section.

Intent-Force Based HMM

A dataset of joint-velocity and grip-force pairs is assumed, collected by executing predefined velocity profiles on a sensorized robotic hand. This dataset captures the relation between velocity inputs and tactile responses and is used to train an HMM that identifies their latent structure. In the following, without loss of generality, a single generic finger is considered i, with $i \in \{1, ..., m\}$, embedding its intent-force correlation structure into a set of discrete hidden states. Let $S^i = \{s_1^i, s_2^i, ..., s_{N_p}^i\}$ denote the set of possible hidden states. At each instant of time, the system observes a multidimensional signal $O^i(t) \in \mathbb{R}^D$, where $O^i(t)$ is a joint observation vector composed of both grasping motion velocity intent $\sigma(t)$ and the Euclidean norm of finger's tactile measurements $||T_i||_2$. These observations are therefore embedded into

hidden-state-dependent multivariate probabilistic distributions: specifically, an observation $O^i(t)$ in state s^i_j ($j \in \{1,...,N_p\}$) is described as a multivariate Gaussian emission probability with a mean vector $\mu^i_j \in \mathbb{R}^D$ and covariance matrix $\Sigma^i_j \in \mathbb{R}^{D\times D}$. In this way, the intent-force correlation is encapsulated in an HMM η^i defined by (Rabiner (1989)): a transition matrix $A^i \in \mathbb{R}^{N_p\times N_p}$ collecting the transition probabilities α^i_{lj} from state s_l to state s_j; the collection of observations' mean $\mu^i = \{\mu^i_j\}_{j=1,...,N_p}$ and covariance $\Sigma^i = \{\Sigma^i_j\}_{j=1,...,N_p}$ for all states, and the collection $\Pi^i = \{\pi^i_j\}_{j=1,...,N_p}$ of initial probabilities of being in state s^i_j. In short: $\eta^i = \{A^i, \mu^i, \Sigma^i, \Pi^i\}$. These model parameters are learned by maximum likelihood estimation via the well known expectation-maximization algorithm (Rabiner (1989)), obtaining the estimate $\hat{\eta}^i$. The goal is to use the HMM $\hat{\eta}^i$ to identify, during HITL control, the relevant regions of the intent–force plane and adapt the interface parameters accordingly. Although such regions could be defined deterministically through empirical rules, they would only represent an idealized approximation of the grasping behavior, which in practice is affected by uncertainties and non-idealities. For this reason, the same ideal regions are encoded within the HMM through appropriate initialization of means, covariances, and transition structure. In A^i, forbidden transitions α^i_{lj} are set to zero, while others are estimated via EM. The initial state distribution Π^i sets $\pi^i_j = 1$ for the chosen starting state s^i_j. For each region $\mathscr{R}^i_j$, the mean μ^i_j is initialized at its centroid, and the covariance Σ^i_j is chosen so that points in $\mathscr{R}^i_j$ attain maximal likelihood under $\mathscr{N}(\mu^i_j, \Sigma^i_j)$, refined through EM. Training then fits the ideal intent–force structure to real tactile data, absorbing hardware-related distortions. Once the trained HMM $\hat{\eta}^i$ is obtained, it becomes therefore possible to infer online – given a running window of M samples of instantaneous intent-force observations O^i – the posterior probability $p^i_j(t)$ of the instantaneous system's state $s^i(t)$ being equal to a particular hidden state s^i_j at each time step, i.e.

$$p^i_j(t) = P(z(t) = s^i_j \mid O^i, \hat{\eta}^i), \quad j = \{1,2,...,N_p\}, \tag{6.7}$$

for the generic finger i, which can be computed through posterior decoding via the forward-backward recursion (Rabiner (1989)). This enables real-time state-based interface modulation, detailed in Sec. 6.3.

Probabilistic Intent Modulation

For each finger i, the posterior probabilities $p_1^i(t), \ldots, p_{N_p}^i(t)$ obtained from eq. (6.7) are used to compute an intent modulation gain

$$\nu^i(t) = \sum_{j=1}^{N_p} \kappa_j \, p_j^i(t), \qquad i \in \{1, \ldots, m\}, \tag{6.8}$$

where each scalar κ_j specifies the modulation associated with state s_j^i. To shape the behavior, two reference values are defined: a larger gain $\kappa_{\max}$, used for states representing free-space motion to allow fast responses, and a smaller gain $\kappa_{\min}$, used for contact states where fine grip force regulation is required. Any remaining states can be assigned intermediate κ_j values to support additional behaviors. In this way, intent scaling naturally adapts to the inferred state and remains independent of the specific robotic hardware. The resulting gain $\nu^i(t)$ scales the user-provided velocity intent $\sigma(t)$ as

$$\tilde{\sigma}^i(t) = \nu^i(t) \, \sigma(t), \qquad i \in \{1, \ldots, m\}, \tag{6.9}$$

providing the diagonal entries of $\Sigma(t)$ in eq. (6.6) and thus the joint velocity command $\dot{q}(t)$ in eq. (6.5).

Grip Force Processing

The grip force level $w(t)$ is estimated from the tactile measurements $\tau(t)$ at the fingertips. The grasp internal forces are approximated (i.e., forces in the null space of the *grasp matrix*, well known from grasping theory, see (Murray (2017))) by exploiting a typical precision tripodal grasp, where the opposing fingers balance the thumb. Under this assumption, the thumb's contact force provides a reliable estimate of the internal force. Thus, given $T_1(t) := T^{\text{thumb}}(t) = [T_x^{\text{thumb}}(t) \; T_y^{\text{thumb}}(t) \; T_z^{\text{thumb}}(t)]^T$, the grip force is computed as

$$w(t) = \| T^{\text{thumb}}(t) \|_2. \tag{6.10}$$

Sensory Feedback Signal Encoding

Once the instantaneous grip force level $w(t)$ is estimated, it is encoded into a vibrotactile feedback signal $\psi(t)$ to be delivered to the user. Specifically, the grip force range $[w_{\min}, w_{\max}]$

is linearly mapped to a normalized feedback signal $\psi(t) \in [0,1]$ according to

$$\psi(t) = \frac{w(t) - w_{\min}}{w_{\max} - w_{\min}}, \tag{6.11}$$

which is then used to modulate the duty cycle of the vibrotactile stimulation through the PWM strategy previously described in Sec. 6.2.1.

6.3 Experimental Validation across Tasks

The experimental validation examined static force regulation, a functional multi-object manipulation task, and a force-informed programming-by-demonstration study, using both an anthropomorphic hand and a parallel-jaw gripper.

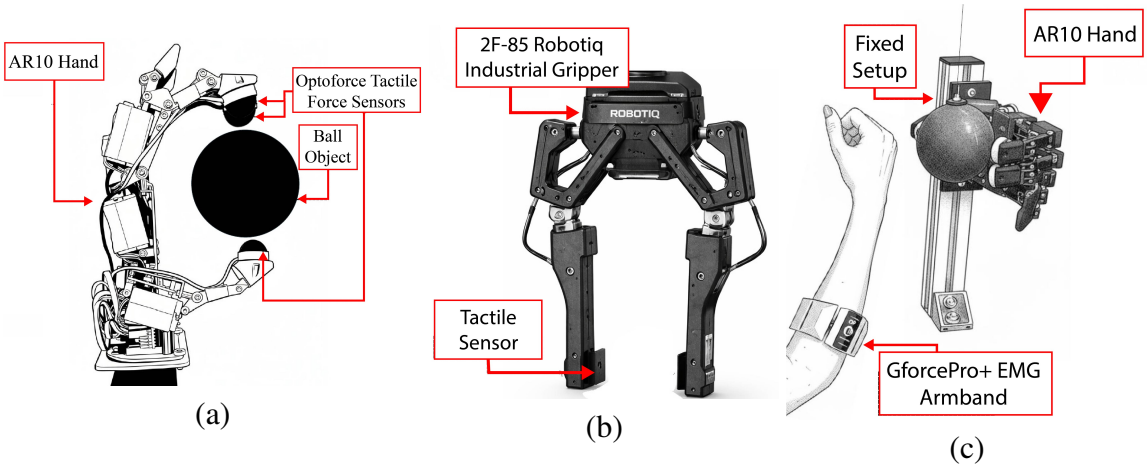


Figure 6.2 (a) AR10 anthropomorphic hand with OptoForce tactile fingertips during grasping. (b) Robotiq 2F-85 parallel gripper with integrated tactile sensing. (c) Static force regulation setup (*Experiment 1*) with sEMG-based control and vibrotactile force feedback.

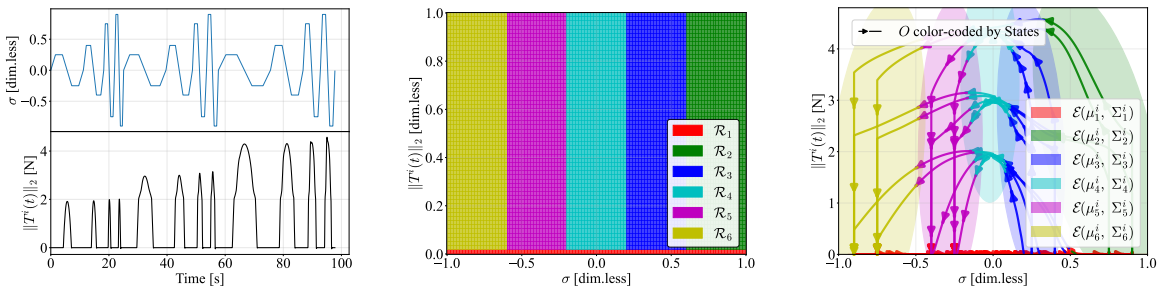
6.3.1 Hardware Platforms and Subjects

Two robotic platforms were used to demonstrate cross-hardware applicability: the AR10 anthropomorphic hand by Activ8 Robots, equipped with tri-axial fingertip force sensors (Fig. 6.2(a)), and the Robotiq 2F-85 parallel-jaw gripper with 5×5 tactile arrays (Fig. 6.2(b)). Ten participants (30 ± 4 years, 3F/7M) were involved, and signed a consent form explaining all details of the experiment. They wore an 8-channel GForce-Pro sEMG armband by OYMotion (Fig. 6.2(c)), sampled at 1 kHz and preprocessed and mapped to a single velocity

intent as detailed previously in Sec. 6.2.1, and joint velocities were generated through synergy control as detailed earlier in Sec. 6.2.2. Grip force feedback was provided through a PWM-driven vibrotactile bracelet worn on the upper arm, using a coin-type vibration motor by Seeed Grove, capable of producing clearly perceptible stimuli; its operation follows the scheme introduced in Sec. 6.2.1, closing the perceptual loop required for fine HITL force regulation.

6.3.2 Hidden Markov Model Training Realization

Prior to real-time deployment, the HMM were trained to capture the latent temporal phases of grasp execution using velocity commands and tactile signals. For each robotic platform (AR10 Hand and Robotiq Gripper), training data were collected by executing grasping motions over three pre-defined, automatic discrete closure levels and four joint velocity profiles, resulting in 12 grasping conditions per device. During data acquisition, both end-effectors were rigidly mounted on a fixed frame and interacted with a spherical object (Fig. 6.2(a)) placed in front of the hand. The sphere was selected as a geometrically neutral object to ensure repeatable fingertip contacts, compatibility with both grasping modalities, and reduced object-dependent contact artifacts, providing a controlled baseline for the training of the HMM.



(a) (b) (c)

Figure 6.3 HMM modeling on the input velocity–force plane: (a) illustration of finger's velocity profile and example of corresponding finger force constituting the dataset for HMM training; (b) regions $\mathscr{R}_j$ of the intent–force plane used for the initialization of the Gaussian emission according to the hidden-state structure of Fig. 6.4; (c) trained model showing the learned Gaussian emissions and exemplifying trajectories colored by the maximum-a-posteriori hidden state in the intent–force plane.

The same protocol was applied to the AR10 hand and the Robotiq gripper to ensure dataset comparability. Both devices were driven in velocity control using synergy-based commands (Sec. 6.2.2) that span the canonical grasp phases. Trapezoidal velocity profiles

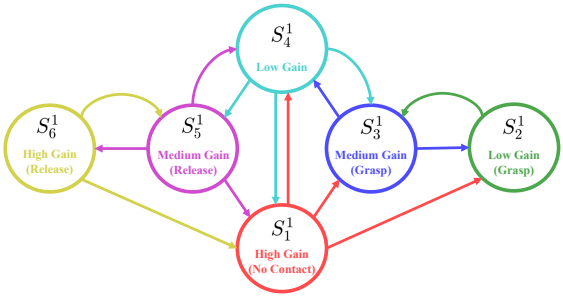


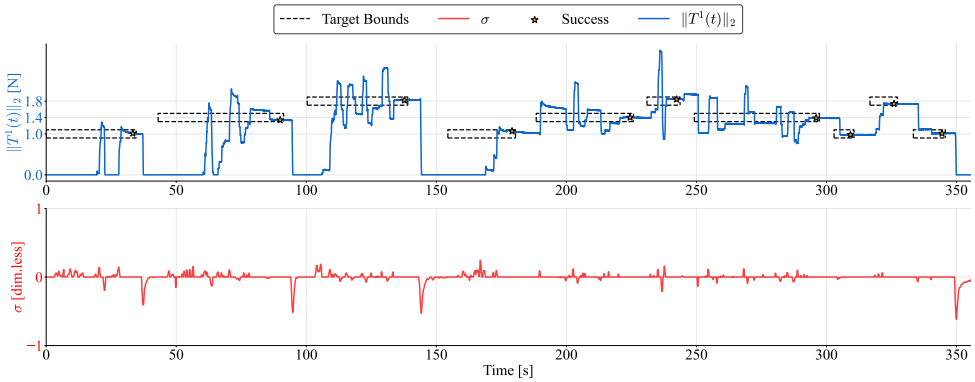
Figure 6.4 Topology of the imposed HMM structure and allowed state transitions. Each node corresponds to a specific gain modality and grasp phase, while arrows indicate allowed transitions, which probabilities are optimized during training.

(Fig. 6.3a) were executed from full opening to contact, force regulation, and full release, with four velocity magnitudes to cover the velocity–force space. Synergy velocities $\tilde{\sigma}^i(t)$ and tactile force magnitudes $\|T^i(t)\|_2$ were recorded and used to train per-finger HMMs (Sec. 6.2.3).

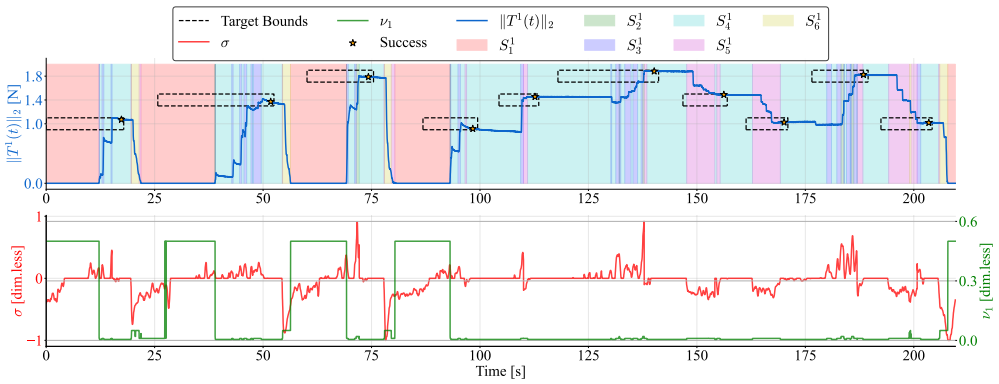
Velocity–Force Plane and Transition Topology of the HMM

For each sensorized finger, the dataset consists of paired observations $O^i(t) = [\,\sigma(t)\;\; \|T^i(t)\|_2\,]^T$ defined on the velocity–force plane, which constitutes the domain of the HMM and where the Gaussian emission probabilities of the hidden states are fitted. The trained model is used online to compute a probabilistic modulation gain that maps the decoded velocity intent $\sigma(t)$ into the effective synergistic command $\tilde{\sigma}^i(t)$, as described in Sec. 6.2.3, eqs. (6.8)–(6.9). To embed structured grasp phase knowledge, a predefined HMM topology was imposed prior to training, fixing the number of hidden states and the admissible transitions, while learning the corresponding probabilities via the Expectation–Maximization algorithm. The resulting sparse transition structure, illustrated in Fig. 6.4, comprises six hidden states capturing the canonical grasp–release cycle on the velocity–force plane. One high-gain state relates to free motion in the absence of contact (red state in Fig. 6.4). Contact conditions are related to two grasp states: a high-gain grasp state (green) and a medium-gain grasp state (blue), followed by a low-gain regulation state (cyan) related to small velocities with nonzero contact forces. Release phases knowledge is captured by two symmetric states, corresponding to medium-gain (violet) and high-gain (yellow) release phases. Each state s_j ($j = 1,\ldots,6$) is associated with a scalar gain coefficient κ_j, with $\kappa_j \in [\kappa_{\min}, \kappa_{\max}]$, where high-gain states are assigned $\kappa_j = \kappa_{\max}$, the low-gain state is assigned $\kappa_j = \kappa_{\min}$, and medium-gain states are associated with intermediate values κ_{mid}. During online operation, the instantaneous modulation gain

is computed as a posterior-probability-weighted combination of κ_j, yielding smooth and phase-dependent scaling of the user's velocity intent (eqs. (6.8)–(6.9)). Initialization of the Gaussian emission parameters plays a crucial role, and was guided by predefined geometric regions $\mathscr{R}_j$ on the $(\sigma, \|T^i\|_2)$ plane, as detailed in the following.



(a) Without shared autonomy

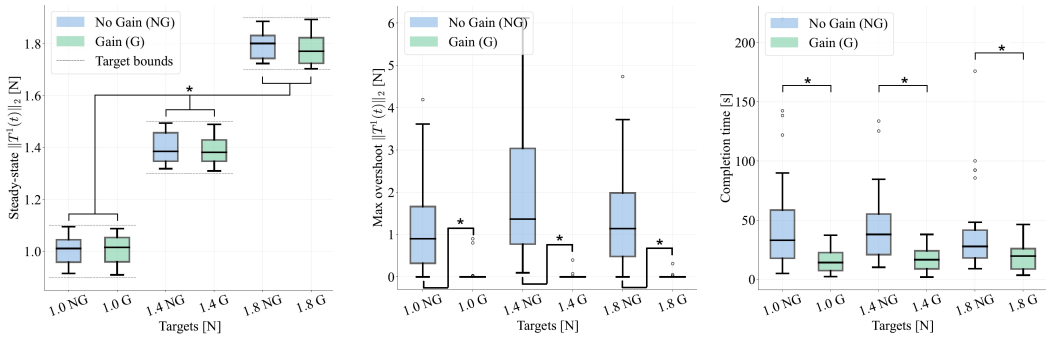


(b) With shared autonomy

Figure 6.5 Static grasp force regulation task (*Experiment 1*) (representative **single subject results**). (a) Without shared autonomy: grip strength (blue) exhibits large oscillations around the target bands (dashed), along with corresponding user intent (red) before convergence. (b) With shared autonomy: grip strength (blue) reaches the target bands smoothly with minimal oscillation during the transient, with background colors indicate the most probable HMM state (consistent with Fig. 6.4), while user intent (red) and instantaneous mapping gain (green) highlight the smoother regulation mechanism.

Initialization of Gaussian Emissions

The initialization of the Gaussian emission distributions was defined according to Fig. 6.3c, which reports the layout of the regions $\mathscr{R}_j$ on the velocity–force plane, colored consistently with the state representation in Fig. 6.4. Each region $\mathscr{R}_j$ represents the nominal domain of a



(a) (b) (c)

Figure 6.6 Static grasp force regulation (*Experiment 1*, **aggregated across subjects**) with (blue) and without (red) shared autonomy. (a) Steady-state target force; (b) absolute force over/undershoot; (c) task completion time. HMM-based shared autonomy reduces variability and over/undershoot, enabling smooth convergence. * $p < .01$.

hidden state and was positioned based on the previously described functional interpretation: the high-gain (no contact) region spans the vicinity of the force-axis origin for both positive and negative velocities. Grasping phases are represented in the first quadrant by the high-gain and medium-gain grasp regions, placed at increasing force levels for large and moderate positive velocities, respectively. The low-gain region occupies a narrow band around small absolute velocities at nonzero force levels, spanning both the first and second quadrants and corresponding to fine force regulation. Symmetrically, medium-gain and high-gain release regions are located at moderate and high negative velocities, respectively. This initialization encodes an idealized intent–force partitioning, which does not explicitly account for mechanical nonidealities such as elastic deformations, frictional hysteresis, and nonlinear compliance of the actuation and transmission mechanisms. As a result, the actual data distribution may deviate from the ideal regions, leading to partial overlap among states. For this reason, the initialization in Fig. 6.3b is interpreted as a structured prior that guides the Expectation–Maximization training procedure. Mean vectors μ_j were initialized at the centroids of the regions $\mathscr{R}_j$, while covariance matrices Σ_j were set so as to maximize the relative likelihood of samples within each region. During training, the Expectation–Maximization algorithm refined both the Gaussian parameters and the allowed transition probabilities, adapting the ideal intent–force structure to the empirical velocity–tactile dataset and yielding well-separated state distributions consistent with the intended gain hierarchy.

Learned Emission Distributions in the Velocity–Force Plane

HMM training outcomes are reported in Fig. 6.3c. Colors are consistent with Fig. 6.4 and identify the same six hidden states and corresponding gain modalities. Differently from the initialization, the colored shapes represent the learned Gaussian emission distributions, visualized as two-dimensional confidence ellipses. For each state s_j, the ellipse is centered at the trained mean $\hat{\mu}_j$ and shaped according to the trained covariance $\hat{\Sigma}_j$, corresponding to the level set $\{x \in \mathbb{R}^2 : (x-\hat{\mu}_j)^\top \hat{\Sigma}_j^{-1}(x-\hat{\mu}_j) = c\}$ for a fixed constant c. The velocity–force trajectories used for training are overlaid as samples and color-coded according to the maximum-a-posteriori state assignment

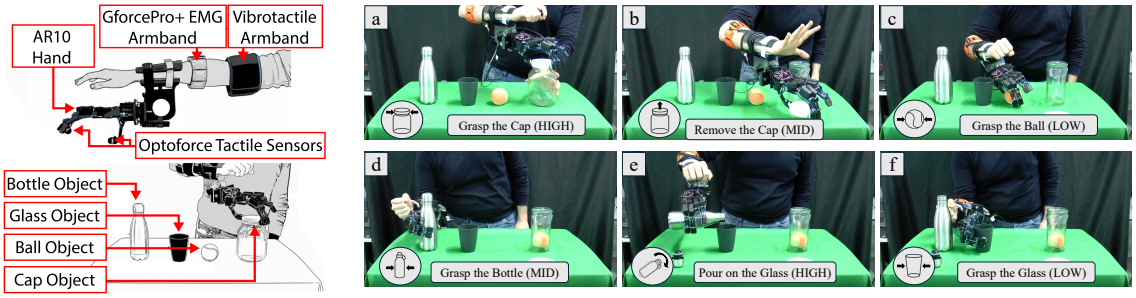
$$\arg\max_j \; p_j(t), \qquad p_j(t) = P\big(z(t) = s_j \,\big|\, O_{1:T}, \hat{\eta}\big), \tag{6.12}$$

where $z(t)$ denotes the hidden state at time t, $O_{1:T}$ is a window of T observations, and $\hat{\eta}$ denotes the set of trained HMM parameters. As shown in Fig. 6.3c, the learned emission distributions preserve the intended functional partition of the velocity–force plane and the associated gain hierarchy, while adapting to nonideal interaction dynamics induced by hardware and contact compliances, uncertainties, nonidealities.

6.3.3 Experiment 1: Systematic Study – Static Grasp Force Regulation

Description

This protocol evaluated the ability of the proposed framework to regulate grip force under static, highly controlled conditions. The robotic hand was rigidly mounted on a fixed frame so that interaction forces depended only on finger motion (Fig. 6.2a), while users controlled closure through sEMG signals. Three reference forces were prescribed: 1.0 N, 1.4 N, and 1.8 N, with a tolerance band of ± 0.1 N. A trial was considered successful when the force entered the band and was maintained for 5 s (*steady-state*). Real-time vibrotactile feedback conveyed grip strength, allowing force regulation without visual inspection. The experiment was designed to isolate the effect of the HMM-based shared autonomy. Two conditions were compared: a baseline condition, in which decoded velocity intent was directly mapped to the synergy controller, and a proposed condition, in which velocity commands were probabilistically modulated by the HMM through the gain described in Sec. 6.2.3. The results of this static evaluation provide a controlled reference for the subsequent functional experiments in Sec. 6.3.4.

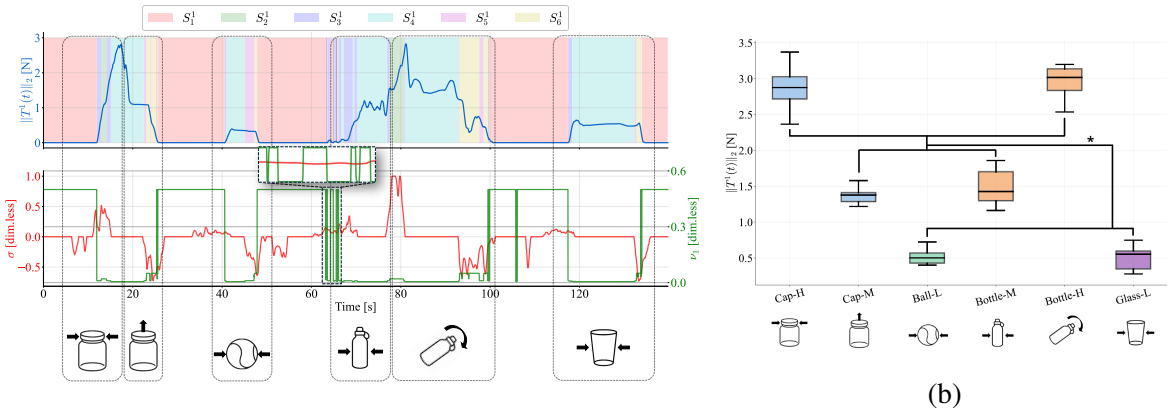


(a) (b)

Figure 6.7 Functional Mock Recipe Preparation task (*Experiment 2*) under HMM-based shared autonomy: (a) experimental setup; (b) example execution sequence during recipe mixing.

Results and Discussion

Figure 6.5a shows a representative trial for one subject without shared autonomy. The upper panel plots the measured grip force with dashed rectangles indicating the target bands; these bands are the same intervals communicated to the user, through the vibrotactile interface, as the target to be achieved. As can be observed, the force trajectories exhibit large oscillations and repeated overshoots before settling within tolerance, revealing the difficulty for participants to compensate nonideal mechanical effects such as stiction, frictional hysteresis, and joint/backdrive compliance when intent is mapped to the robot hand command by means of a fixed gain. The lower panel reports the user's intent, which alternates bursts of opposite sign as the participant attempts to counteract the nonlinearities and drive the force back into bounds, often resulting in additional overshoot. Conversely, when the HMM-based shared autonomy is enabled, the behavior changes significantly, as can be observed in Fig. 6.5b. In the top panel, the background color encodes, at each instant, the HMM state with higher probability, using the same color scheme introduced in Fig. 6.4. Transitions from the free-motion state to grasp states occur at contact; during force buildup the model progresses through medium/high grasp gains and, as the target is approached, naturally moves into the low-gain state that enlarges control resolution. The resulting force trajectories rise smoothly and settle reliably within the target band with minimal ringing. The bottom panel reports both the user's velocity intent and the instantaneous gain applied to the thumb synergy. Consistently with the state sequence, the gain takes high values during free motion, drops to low or medium levels in contact for fine regulation, and varies only modestly while the user makes small intent adjustments to stay within tolerance. This closed-loop probabilistic modulation is precisely what suppresses the overshoot–correction cycles observed in the baseline.



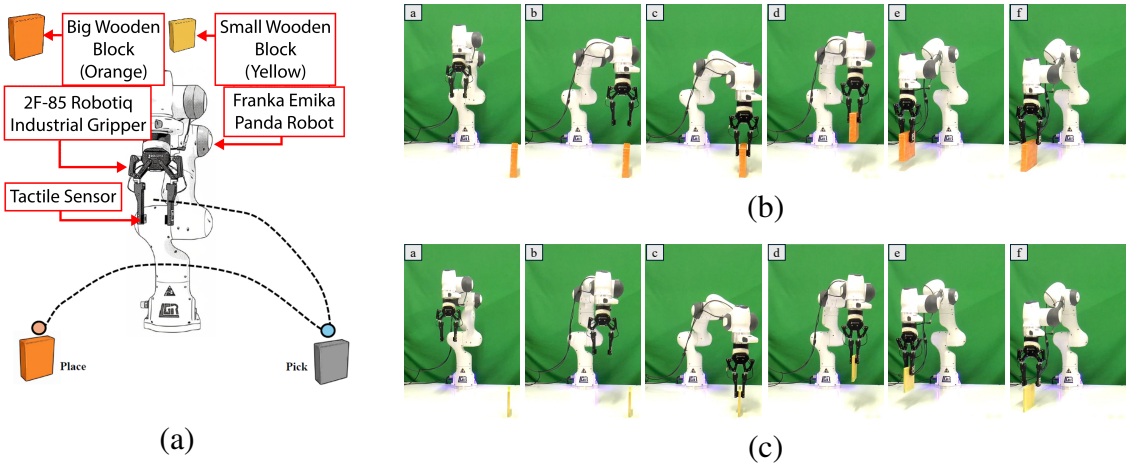
(a) (b)

Figure 6.8 Functional Mock Recipe Preparation task (*Experiment 2*) under HMM-based shared autonomy: (a) single-subject execution of the task: the top graph reports grip force (blue) along with colored areas indicating the HMM state with instantaneous highest probability, while the bottom graph shows user intent modulation (red) and shaping of the mapping gain (green); (b) aggregated results for all subjects: grip forces achieved across objects according to user naturally-regulatated LOW, MEDIUM and HIGH strength levels. * indicates statistically significant difference ($p < .01$).

Then, aggregate results across all participants are summarized with boxplots. Figure 6.6a reports the attained steady-state forces for each target with and without shared autonomy. As expected, both conditions reach the prescribed levels, confirming that the shared autonomy does not bias the attainable steady force. A one-way ANOVA reveals a significant effect of the control condition for all targets (1.0 N: $F_{1,18} = 15.42$, $p < 0.01$; 1.4 N: $F_{1,18} = 18.76$, $p < 0.01$; 1.8 N: $F_{1,18} = 21.33$, $p < 0.01$). The difference emerges in transient quality. Figure 6.6b quantifies the peak overshoot relative to the target band. Without shared autonomy the overshoot distribution is markedly larger and more dispersed across all targets, indicating unreliable transients and frequent overcorrections; instead, with the HMM-based probabilistic gain, overshoot remains consistently small. This effect is confirmed by one-way ANOVA, showing significantly lower overshoot with shared autonomy for all targets (1.0 N: $F_{1,18} = 27.85$, $p < 0.01$; 1.4 N: $F_{1,18} = 31.12$, $p < 0.01$; 1.8 N: $F_{1,18} = 24.67$, $p < 0.01$). Overall, these results validate the benefiting role of the probabilistic model in a systematic manner in controlled conditions. By interpreting velocity-force observations through the HMM and modulating the velocity-to-synergy mapping accordingly, the controller increases responsiveness in free motion, automatically lowers gain for fine force regulation upon contact, and restores higher gain for release. The shared autonomy improves reliability, reduces overshoot, and shortens completion time without sacrificing the attainable force

level, as further confirmed by one-way ANOVA on completion time (1.0 N: $F_{1,18} = 19.08$, $p < 0.01$; 1.4 N: $F_{1,18} = 22.94$, $p < 0.01$; 1.8 N: $F_{1,18} = 17.61$, $p < 0.01$).

6.3.4 Experiment 2: Functional Task – Mock Recipe Preparation



(a) (b) (c)

Figure 6.9 Tactile-informed manipulation via PbD task. (a) Experimental setup with the Franka Emika Panda Robot and the Robotiq 2F-85 industrial gripper; (b) manipulation sequence with the orange *big* block; (c) manipulation sequence with the yellow *small* block.

Description

Building on the controlled static analysis of *Experiment 1* (Sec. 6.3.3), this experiment evaluates whether the proposed HMM-based shared autonomy can be effectively exploited in a functional and dynamically rich task. Rather than re-testing individual control components, the goal is to verify that the previously characterized capabilities translate into successful task execution in a realistic scenario. *Experiment 2* is a goal-oriented task inspired by everyday recipe preparation, requiring grip-force regulation across objects with different functional demands. The task was performed using the AR10 anthropomorphic hand with OptoForce fingertip sensors, controlled via sEMG-derived velocity intent and guided exclusively by vibrotactile feedback (Secs. 6.3.1, 6.2.1). As shown in Fig. 6.7a, the hand was worn by participants through a custom forearm-mounted support, allowing natural arm motion while grasp control remained sEMG-driven. The task required transitions between LOW, MEDIUM, and HIGH grip force levels, defined through a vibrotactile reference range rather than fixed thresholds. A maximum vibration level corresponding to the HIGH force condition was provided, while LOW and MEDIUM forces were achieved by natural modulation within

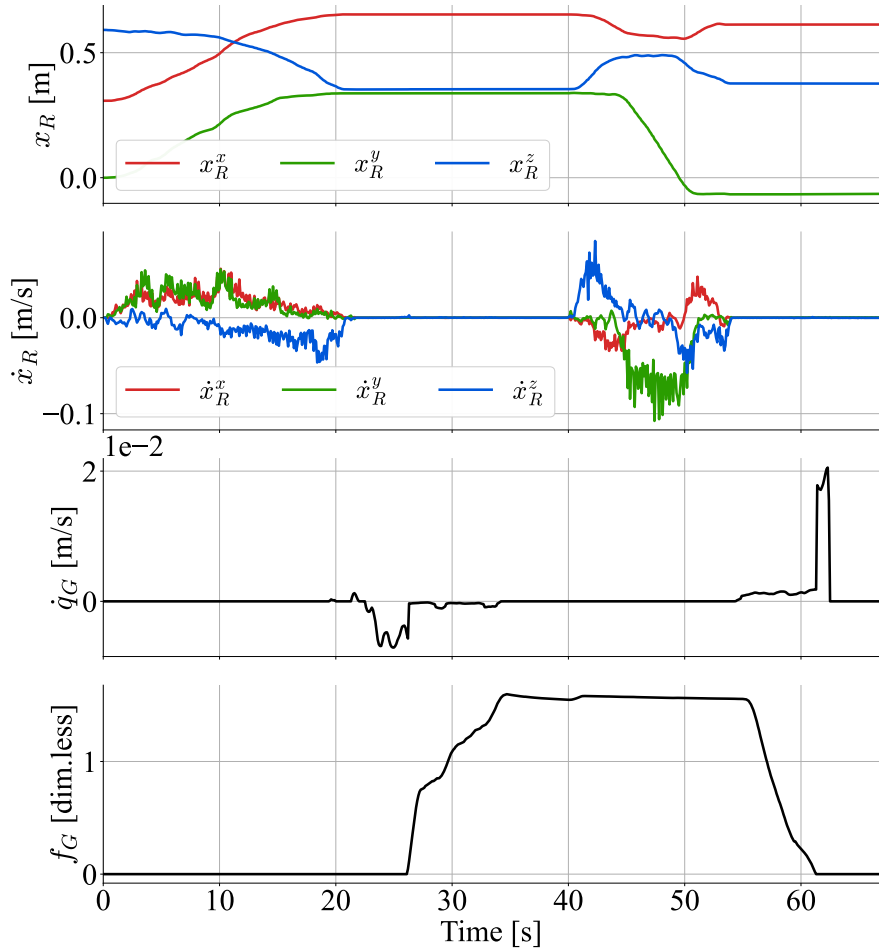


Figure 6.10 Example of the multi-modal demonstration signals used for HMM training. The plots show end-effector position and velocity, gripper velocity, and the demonstrated grip force $f_G(t)$. The temporal structure of the task (approach, contact, grasp tightening, transport, release) is clearly visible in both the motion and force channels.

the perceived range. This design assessed task-driven force regulation using the full human–robot control loop. The sequence of actions included: unscrewing a jar cap with HIGH force, transporting it with MEDIUM force, placing a lightweight ball using LOW force, handling a bottle with MEDIUM force, pouring with HIGH force, and finally grasping and releasing a plastic glass with LOW force. A trial was considered failed if an object slipped or fell, or if grip force exceeded 3.5 N. No failures were observed, indicating effective use of vibrotactile feedback and HMM-based shared autonomy throughout the task. Fig. 6.7b shows representative task snapshots.

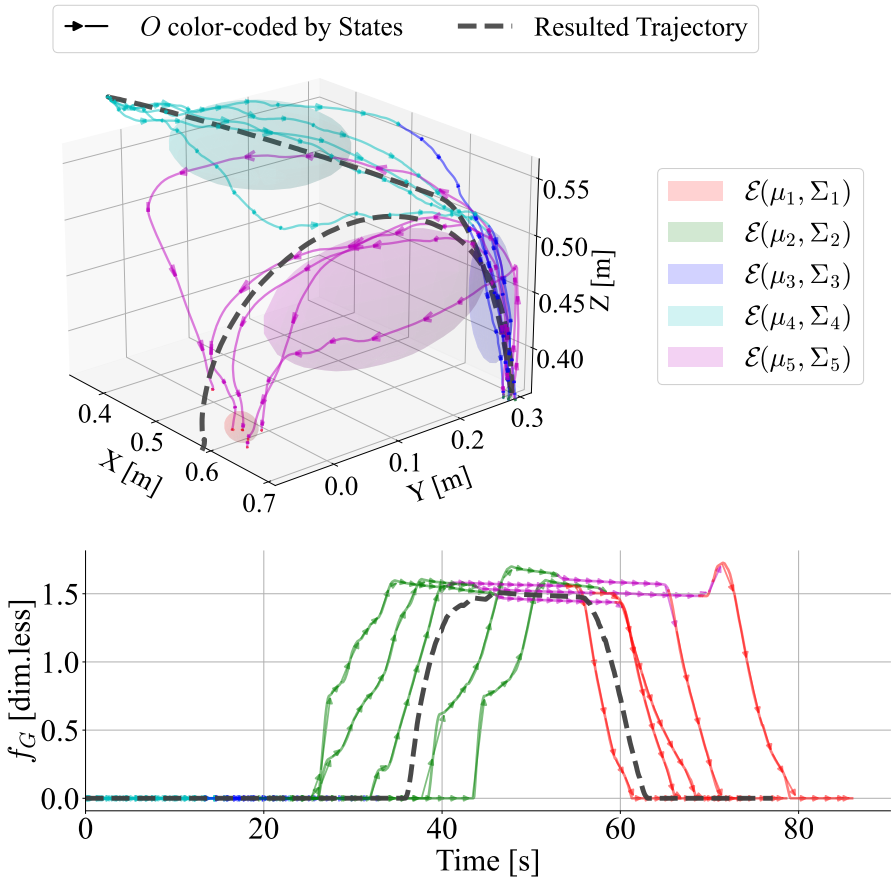


Figure 6.11 Top: online automatic execution of the manipulation task (dashed), along with user-demonstrated trajectories represented in the state space of the higher level PbD-HMM, with colors indicating higher probability state along time. Bottom: online automatic profile of the grip force (dashed), along with user-demonstrated grip force profiles $f_G(t)$ used to train the force component of the model.

Results and Discussion

Figure 6.8 provides a summary of the mock recipe preparation task under the HMM-based shared-autonomy condition, including a representative single-subject execution and the aggregated grip-force statistics across all participants. Fig. 6.8a illustrates the temporal evolution of the task for a representative subject. The plot shows how grip force is continuously modulated over time in response to the sequential task requirements, with transitions between LOW, MEDIUM, and HIGH functional force levels as different objects are manipulated. The shaded regions denote the task phases associated with each object interaction, while the force trajectory shows gradual adjustments without large overshoots in this representative trial. In particular, grip force increases during actions requiring firmer grasps, as required by the

protocol for jar and bottle-cap unscrewing, and is subsequently reduced for more delicate manipulations, including ball and glass handling. Throughout the illustrated execution, the applied force remains within the predefined 3.5 N upper safety threshold.

On the other hand, Fig. 6.8b reports the distribution of measured grip forces achieved across all participants for each object–grip-level pair. The boxplots show a clear separation among the LOW, MEDIUM, and HIGH functional force levels prescribed by the task. This separation is statistically confirmed by a one-way ANOVA across force levels, revealing a significant main effect of the functional grip level (LOW vs MEDIUM vs HIGH, $F_{2,57} = 86.41$, $p < 0.01$). Specifically, HIGH-force conditions (Cap-H, Bottle-H) differ significantly from MEDIUM-force conditions (Cap-M, Bottle-M), which in turn differ significantly from LOW-force conditions (Ball-L, Glass-L) ($p < 0.01$). More concretely, the jar cap was grasped using distinct HIGH and MEDIUM force levels in accordance with the protocol requirements; the ball was manipulated at LOW force; the bottle exhibited medium forces during unscrewing and higher forces during pouring; and the glass was handled at LOW force across all subjects. No condition shows force values exceeding the 3.5 N upper safety bound. Taken together, these results support the narrower claim that, in this multi-step functional task, users were able to differentiate force levels while the probabilistic shared-autonomy mechanism helped maintain user control and keep the applied forces within the studied range.

6.3.5 Force-Aware Programming by Demonstration

Beyond real-time human-in-the-loop control, the proposed HMM-based shared autonomy enables grip-force-aware skill acquisition via Programming by Demonstration (PbD). In this experiment, it is investigated whether the same probabilistic modulation mechanism used to support users during manipulation can also serve as a teaching interface, allowing a human operator to demonstrate not only motion trajectories but also task-relevant grasp-force profiles. The goal is to encode such demonstrations into a hierarchical HMM (H-HMM) that captures the coupled evolution of motion and grip strength, enabling autonomous reproduction of tactile-informed manipulation skills. From a structural standpoint, the hierarchy separates two temporal abstractions: low-level contact phases governing intent–force modulation and high-level task phases governing manipulation sequence evolution. This separation is what turns the framework from a pure regulation strategy into a programming architecture.

Motion and Grip-Force Demonstrations

Participants carried out kinesthetic demonstrations of a pick-and-place task using a Franka Emika Panda robot equipped with the Robotiq 2F-85 parallel gripper described in Sec. 6.3.1 (see Fig. 6.9a). During demonstrations, the operator physically guided the robot end-effector while regulating the gripper force through the same vibrotactile-guided HMM-based shared autonomy interface described in Sec. 6.2.3 and validated in Experiments 1 and 2. In this way, grasp force was naturally programmed through the shared-autonomy mechanism. At each time step t, a demonstration sample was defined as

$$y(t) = \begin{bmatrix} x_R(t) \\ \dot{x}_R(t) \\ \dot{q}_G(t) \\ f_G(t) \end{bmatrix}^T , \tag{6.13}$$

where x_R and $\dot{x}_R$ denote the Cartesian position and velocity of the robot end-effector, $\dot{q}_G$ is the gripper velocity command, and f_G is the demonstrated grip strength. Fig. 6.10 reports a representative demonstration, showing the synchronized evolution of motion and force signals. Multiple demonstrations were recorded, each exhibiting a consistent temporal structure composed of approach, contact, grasp tightening, object transport, and release.

HMM Encoding of Motion and Force

The demonstration dataset was modeled with an HMM to jointly encode motion and force information. For each HMM state s_i, a Gaussian emission distribution with mean $\mu_i^{(\cdot)}$ and covariance $\Sigma_i^{(\cdot\cdot)}$ was defined over the combined observation space:

$$\mu_i = \begin{bmatrix} \mu_i^{x_R} \\ \mu_i^{\dot{x}_R} \\ \mu_i^{\dot{q}_G} \\ \mu_i^{f_G} \end{bmatrix}, \qquad \Sigma_i = \begin{bmatrix} \Sigma_i^{x_R} & \cdots & \Sigma_i^{x_R f_G} \\ \vdots & \ddots & \vdots \\ \Sigma_i^{f_G x_R} & \cdots & \Sigma_i^{f_G} \end{bmatrix}. \tag{6.14}$$

This formulation represents the marginal statistics and cross-covariances of each variable, explicitly capturing correlations between motion phases and grip strength. As shown in Fig. 6.11, the resulting HMM states align with probabilistically consistent manipulation subskills, properly associated with characteristic force values.

Reproduction and Generalization to a Different-Sized Object

During reproduction, the learned manipulation skill was executed autonomously by the robot using the parameters encoded in the HMM. Gaussian Mixture Regression (GMR) (Calinon et al. (2010)) was employed to generate the desired end-effector velocity $\hat{\dot{x}}_R$ conditioned on the inferred task phase, while the corresponding grip force profile was retrieved from the learned force component $\mu_i^{f_G}$ of the active HMM state. This enabled the robot to reproduce both motion and force modulation without any further human intervention. The reproduction results are illustrated in Fig. 6.11. In particular, the black dashed line in the figure represents the end-effector trajectory autonomously reproduced by the robot based on the learned HMM. The color-coded segments indicate the inferred HMM states along the execution, highlighting the alignment between motion phases and the associated grasp force behavior. Furthermore, to assess limited transfer capabilities, the learned task was reproduced using two objects of different sizes: the same wooden block used during demonstrations and a thinner block placed at the same location. Despite the reduction in object thickness, the reproduced end-effector trajectory remained unchanged, as evidenced by the identical black dashed trajectory shown in Fig. 6.11 for both objects. This result indicates that the learned motion plan can be preserved across the two tested object sizes, while the grasp execution adapts to the task requirements. At the same time, the grip-force profiles associated with the execution adjusted to the object properties while remaining consistent with the demonstrated teaching pattern. Overall, these results show that *(i)* the demonstrated force profiles were captured and encoded within the H-HMM, *(ii)* the learned skill transferred to a second object size within the tested setup without retraining, and *(iii)* motion and force components were decoupled at execution time, allowing the robot to preserve the task-level trajectory while adapting grasp force as needed. This experiment therefore supports the use of the proposed HMM-based shared-autonomy interface as a tactile-informed teaching interface for force-aware manipulation in the studied setting.

6.4 Discussion on Generalization and Autonomy Scaling

A probabilistic shared-autonomy framework for grip-force-aware human-in-the-loop control and programming of robotic grasping has been presented using sEMG and tactile sensing. By modeling the coupled relationship between user intent and contact force through HMMs, the approach enables adaptive modulation of user commands, allowing fast free-space motion and precise force regulation upon contact. Systematic experiments under controlled static

conditions showed lower overshoot and more reliable force-level differentiation than direct intent mapping in the reported conditions. Functional manipulation experiments further showed that users could exploit the shared-autonomy mechanism to regulate grip force across multi-object, task-oriented scenarios without explicit force targets. Finally, a programming-by-demonstration pilot study showed that the same interface can also encode force-informed manipulation skills, enabling autonomous reproduction and limited transfer to objects of different sizes while preserving task-level motion structure.

Within the dissertation, this chapter therefore extends the control-oriented contribution of Chapter 5 toward skill-level encoding. At the same time, the present evidence remains bounded: the programming-by-demonstration validation is still task-specific, the number of demonstrated skills is limited, and transfer is only shown for modest object-size variation. The chapter should therefore be interpreted as evidence that probabilistic shared autonomy can support force-aware skill encoding in the tested setting, rather than as a general solution for arbitrary dexterous manipulation.

Chapter 7

Phase-Dependent Probabilistic Latent Motion Representation

7.1 Limitations of Global Synergy Models

While Chapters 5 and 6 addressed uncertainty during force regulation and skill execution, this chapter focuses on uncertainty in the representation of reach-to-grasp motion itself. The central question is not how to modulate control authority online, but how to encode phase-dependent hand trajectories so that reconstruction remains accurate when the motion manifold is nonlinear and time-varying. In that sense, the chapter contributes the probabilistic-inference side of the dissertation more directly than the control side. Its main limitation is that the available datasets are still relatively small and task-specific, so the conclusions should be interpreted within that validity domain.

The human hand combines a complex musculoskeletal architecture with exceptional dexterity, making it a canonical model of versatile manipulation required for daily tasks. This has motivated extensive research on representing and controlling grasping motions in robotic hands, including mapping, planning, and control (Ciocarlie and Allen (2010); Meattini et al. (2023b); Shukla and Billard (2012)). Yet reproducing this capability in robots remains difficult because their high degrees of freedom (DoF) dramatically increase the complexity of control, planning, and design (Ciocarlie and Allen (2009); Meattini et al. (2023b)).

Neuroscience indicates that the hand manages this redundancy by exploiting postural synergies, i.e., coordinated patterns of finger joint motion that reduce the effective dimensionality of control (Santello et al. (2016)). Such correlations enable compact low-dimensional representations that retain the essential structure of hand postures. It has been shown that most of the variance in static grasp postures can be captured by only a few principal compo-

nents (PCs) obtained through Principal Component Analysis (PCA) (Santello et al. (1998)). This insight has deeply influenced robotics: by leveraging synergies, grasp movements can be encoded in a low-dimensional space, simplifying control. Building on this principle, synergy-based approaches have enabled efficient grasp planning, mechanically coupled hand designs, and intuitive control strategies for robotic hands (Palli et al. (2014); Santina et al. (2018); Si et al. (2024)).

While PCA-based synergies work well for static grasp postures, their extension to dynamic and nonlinear reach-to-grasp motions is still limited. Grasping is inherently time-varying, evolving from an initial open-hand configuration to a final stable grasp, and accurately representing this evolution is crucial for human-like robotic manipulation (Mason et al. (2001)). However, most synergy-based methods focus on static postures and ignore reach-to-grasp trajectories in the synergy subspace, making them unable to capture the nonlinear and time-varying coordination patterns typical of human grasping. Moreover, the linear nature of PCA limits its ability to model the nonlinear structure of the grasp manifold. Nonlinear dimensionality reduction techniques, such as Gaussian Process Latent Variable Model (GP-LVM), Kernel PCA, and autoencoders, are better suited to capturing curved manifolds and spatio-temporal grasp trajectories (Moon et al. (2024); Romero et al. (2010); Starke et al. (2018)). Nevertheless, when a single global nonlinear latent representation is used to encode complete reach-to-grasp motions, reconstruction accuracy degrades as the diversity of grasp types and trajectories increases. This indicates that global nonlinear synergy representations are insufficient to faithfully model complex, time-varying reach-to-grasp trajectories. In fact, global latent models implicitly assume a quasi-stationary trajectory manifold, which is incompatible with phase-dependent motion structure.

Reach-to-grasp motions exhibit strong locality, with different portions of the trajectory characterized by distinct coordination patterns. Exploiting this locality reduces interference between heterogeneous motion segments and improves both representational accuracy and reconstruction fidelity. Motivated by this observation, this chapter proposes a method based on HMM-driven segmentation and connection of local trajectory nonlinear synergy representations to improve reach-to-grasp modeling and reconstruction. The approach combines probabilistic temporal segmentation, local nonlinear synergy subspace modeling, and probabilistic trajectory blending and reconstruction. A left-right Hidden Markov Model (HMM) probabilistically decomposes grasp trajectories into locally coherent segments, and for each local sub-dataset a back-constrained GP-LVM is learned to capture a local nonlinear synergy representation in a low-dimensional latent space. During reconstruction, the complete reach-to-grasp motion is obtained by probabilistically blending the segment-wise reconstructions

using HMM posterior probabilities, ensuring smooth and temporally consistent transitions. Under this perspective, the proposed representation is not only trajectory-aware but also explicitly phase-dependent.

7.2 Related Work on Synergy Representations

Postural synergy extraction has long been recognized as an effective way to reduce the dimensionality of human and robotic hand control. Linear methods, especially PCA, are widely used for their simplicity and computational efficiency, not only in robotic hand grasping (Katyara et al. (2023)) but also in applications such as exoskeleton robot control (Hassan et al. (2018)) and quadruped locomotion (Stella et al. (2025)). Using only the first two or three PCs typically explains more than 80% of the total variance in grasp data. It has been shown that three PCs capturing over 85% of the variance are sufficient to reconstruct a wide range of static postures on the UB Hand IV (Ficuciello et al. (2012)). However, discarding the remaining PCs inevitably loses information and degrades reconstruction accuracy. Moreover, grasping is not limited to static configurations but evolves continuously from an initial open-hand posture to a final stable grasp. To approximate this process, PCA-based synergies were extended to dynamic motions by linearly interpolating the coefficients between the open and final configurations in the synergy subspace (Ficuciello et al. (2014)). It was further shown that applying PCA directly to complete reach-to-grasp trajectories significantly increases reconstruction error, highlighting the limitations of purely linear synergy representations for dynamic grasping (Xu et al. (2016)).

Since PCA is fundamentally linear and most implementations keep only two or three PCs, reconstructing full grasp trajectories with PCA alone often yields substantial errors, such as premature object contact or inaccurate final postures (Santello et al. (2002)). To alleviate these limitations, PCA-based synergies are frequently combined with complementary techniques. For example, PCA-derived static postures have been used as preshaping configurations and then refined via closed-loop feedback from contact forces and finger positions (Ficuciello (2019)). PCA synergy coefficients have been adapted using a multilevel neural network combined with a reinforcement learning algorithm to achieve rapid convergence to optimal grasps (Monforte and Ficuciello (2021)). Grasp force synergies have been extracted via PCA and encoded with recurrent neural networks to generate human-like force patterns (Starke et al. (2021)). Similarly, PCA-derived synergies have served as initial approximations refined by a synergy-based reinforcement learning policy, ultimately producing stable and functional grasps (Li et al. (2025)).

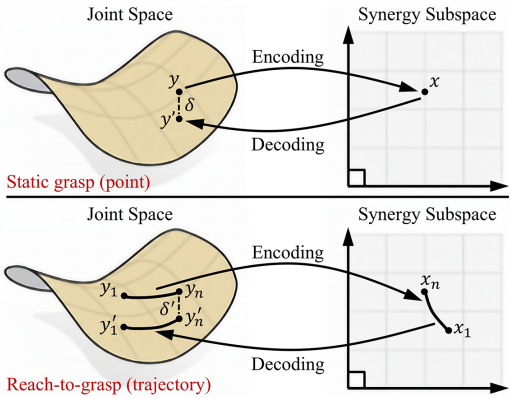


Figure 7.1 Illustration of reconstruction errors for static grasp poses and complete reach-to-grasp trajectories mapped between joint space and synergy subspace.

In contrast to linear approaches, nonlinear dimensionality reduction aims to capture the curved manifolds underlying hand motion and has been shown to improve reconstruction accuracy for high-dimensional grasp postures from low-dimensional latent spaces. It has been demonstrated that GP-LVM can closely reproduce highly nonlinear training trajectories and significantly outperforms both linear interpolation and PCA in reconstructing complete reach-to-grasp motions, on both training and test sets (Romero et al. (2013)). By explicitly modeling the nonlinear structure of reach-to-grasp motion, GP-LVM can faithfully reconstruct full spatio-temporal sequences. Other nonlinear approaches, such as neural network-based autoencoders, have likewise been reported to achieve superior reconstruction accuracy compared to PCA (Starke et al. (2020)). Nevertheless, when a single global nonlinear representation encodes entire reach-to-grasp trajectories, performance degrades as the diversity of grasp types and trajectories increases (Xu et al. (2016)). These limitations indicate that global synergy representations, even when nonlinear, are insufficient to capture the complex, time-varying structure of diverse reach-to-grasp motions.

In summary, dimensionality reduction followed by reconstruction inevitably introduces errors, as illustrated in Fig. 7.1. For the final grasp pose, this effect is relatively limited because reconstruction is not influenced by intermediate trajectory states, allowing even linear methods such as PCA to achieve reasonable accuracy. In contrast, reconstructing complete reach-to-grasp trajectories involves many more data points and must account for the inherently nonlinear and time-varying nature of the motion, leading to substantially higher reconstruction errors. Even nonlinear methods such as GP-LVM show increased final-pose deviation (with $\delta' > \delta$) and noticeable inaccuracies along the full spatio-temporal trajectory, which constrains the synergy representations and the complete reach-to-grasp trajectory reconstructions.

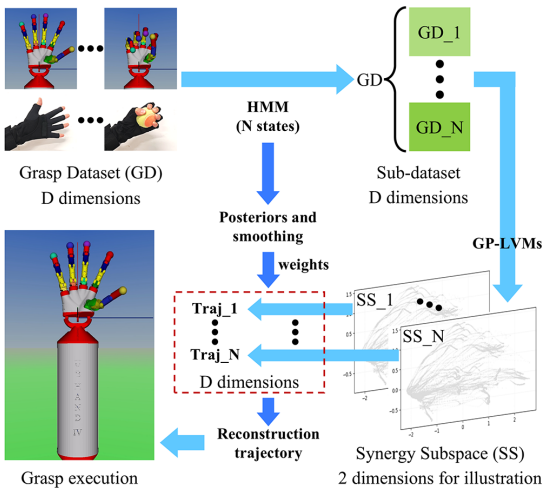


Figure 7.2 Overview of local probabilistic nonlinear latent representation and complete reach-to-grasp trajectory reconstruction.

7.3 Local Probabilistic Latent Representation Framework

7.3.1 Method Overview

As shown in Fig. 7.2, the proposed method relies on HMM-driven segmentation and the connection of local nonlinear synergy representations to improve reach-to-grasp trajectories in the synergy subspace. The framework comprises three stages:

Probabilistic segmentation

A left-right HMM is trained on joint-angle demonstrations to infer latent states along each reach-to-grasp trajectory. Rather than imposing semantically predefined phases, the HMM provides a data-driven probabilistic segmentation that identifies locally coherent regions of the trajectory. Samples assigned to the same latent state across demonstrations are grouped to form local sub-datasets, each corresponding to a statistically homogeneous portion of the reach-to-grasp motion.

Local synergy subspace modeling

For each local sub-dataset, a back-constrained GP-LVM is trained to learn a nonlinear latent representation of the hand motion within that region. These local synergy subspaces capture nonlinear finger coordination patterns with higher fidelity than a single global latent representation, enabling more accurate modeling of time-varying grasp trajectories.

Probabilistic trajectory blending and reconstruction

During reconstruction, each local GP-LVM decodes a complete joint-space trajectory. The final reach-to-grasp motion is produced by probabilistically blending these local reconstructions over time using HMM state posterior probabilities combined with temporal Gaussian smoothing. This blending strategy ensures smooth transitions between local representations and yields a continuous, temporally consistent reconstruction of the reach-to-grasp motion.

7.3.2 Probabilistic Segmentation of Reach-to-Grasp Trajectories

To decompose reach-to-grasp motions into locally coherent trajectory segments, a probabilistic segmentation based on HMM is employed. HMM is well suited to sequential, high-dimensional time-series data with hidden state structure, such as joint-angle trajectories in robotic hand grasping. By modeling the stochastic process that generates observed motion sequences, HMM captures both temporal regularities and transitions between locally homogeneous motion regions across multiple demonstrations.

An HMM with N states is defined by the parameter tuple $\lambda = (\Pi, A, B)$, where:

- $\Pi = \{\pi_n\}_{n=1}^{N}$ denotes the initial state probability distribution, where $\pi_n = P(s_1 = n)$ is the probability that the sequence begins in hidden state n;
- $A = \{a_{ij}\}$ is the state transition matrix, where $a_{ij} = P(s_{t+1} = j | s_t = i)$ represents the probability of transitioning from state i to state j;
- $B = \{b_n(y_t)\}$ defines the observation likelihoods, where $b_n(y_t) = P(y_t | s_t = n)$ is the probability of observing y_t when the system is in hidden state n.

Here, each observation $y_t \in \mathbb{R}^D$ is the D-dimensional joint angle vector of the robotic hand at time step t. The HMM parameters are learned from the grasp demonstrations using the Baum-Welch algorithm (Rabiner (1989)). To reflect the progressive nature of reach-to-grasp motions, which evolve monotonically from an initial open-hand configuration to a final grasp, a left-right Gaussian HMM is adopted. In this structure, state transitions are constrained to remain in the same state or advance to the next state (e.g., $a_{i,i}$, $a_{i,i+1}$), thereby enforcing a consistent temporal ordering along the reach-to-grasp trajectory.

After training, each demonstration is probabilistically segmented according to the inferred hidden state sequence. Trajectory samples associated with the same latent state across all demonstrations are grouped to form local sub-datasets. Each sub-dataset represents a statistically homogeneous portion of the reach-to-grasp motion rather than a semantically

predefined phase. This local segmentation enables subsequent modeling stages, such as nonlinear synergy subspace learning and trajectory reconstruction, to focus on locally consistent motion patterns, thereby improving representation accuracy and reconstruction fidelity.

7.3.3 Local Synergy Subspace Modeling

Following the HMM-based probabilistic segmentation of reach-to-grasp trajectories, each local sub-dataset is modeled independently using a GP-LVM to learn a local nonlinear synergy subspace. GP-LVM is a probabilistic nonlinear dimensionality reduction method that learns a smooth mapping from a low-dimensional latent space to the high-dimensional observation space. In contrast to PCA, which is limited to global linear projections, GP-LVM can capture the highly nonlinear manifolds that characterize coordinated multi-joint hand motions.

Let $\mathbf{Y} \in \mathbb{R}^{T\times D}$ denote the high-dimensional joint-angle observations over T time steps and D joints. The corresponding low-dimensional latent variables are collected in $\mathbf{X} \in \mathbb{R}^{T\times q}$, with $q \ll D$. GP-LVM assumes that each observation $y_i \in \mathbb{R}^D$ is generated from a latent variable $x_i \in \mathbb{R}^q$ through a latent function $f(\cdot)$:

$$y_i = f(x_i) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \tag{7.1}$$

A Gaussian process (GP) prior is placed over each output dimension of $f(\cdot)$:

$$f(\cdot) \sim \mathcal{GP}\big(0, k(\cdot,\cdot)\big), \tag{7.2}$$

where $k(\cdot,\cdot)$ is a covariance kernel function that encodes the smoothness and structure of the mapping. Here, a radial basis function (RBF) kernel is employed:

$$k(x_i,x_j) = \sigma_r^2 \exp\Big(-\frac{\gamma}{2}(x_i-x_j)^\top(x_i-x_j)\Big). \tag{7.3}$$

By integrating out the latent functions, the marginal likelihood of the model can be written as

$$P(\mathbf{Y} \mid \mathbf{X}, \theta) = \prod_{d=1}^{D} \mathcal{N}\big(y^{(d)} \mid \mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}\big), \tag{7.4}$$

where $y^{(d)}$ is the d-th column of $\mathbf{Y}$ and $\mathbf{K}$ is the kernel matrix with entries $K_{ij} = k(x_i,x_j)$. The latent representation $\mathbf{X}$ and kernel hyperparameters θ are obtained by minimizing the

negative log marginal likelihood using gradient-based optimization, with PCA-derived latent coordinates as initialization to improve convergence and avoid poor local minima.

Standard GP-LVM tends to preserve dissimilarities in the data space but does not explicitly guarantee that local neighborhoods are maintained in the latent space. To enforce locality preservation, a back-constraint is employed, in which latent points are computed through a smooth parametric mapping $g_{\mathbf{w}} : \mathbb{R}^D \to \mathbb{R}^q$:

$$x_i = g_{\mathbf{w}}(y_i). \tag{7.5}$$

In the proposed framework, $g_{\mathbf{w}}(\cdot)$ is parameterized by a multi-layer perceptron (MLP) with two hidden layers and hyperbolic tangent activation functions. The network weights $\mathbf{w}$ are learned jointly with the GP-LVM parameters by maximizing the constrained marginal likelihood. The back-constraint ensures that temporally or spatially similar grasp configurations in the original joint space are mapped to nearby locations in the latent space.

When applied independently to each HMM-derived local sub-dataset, the back-constrained GP-LVM yields a compact nonlinear synergy subspace that preserves the essential structure of the original high-dimensional motion. Collectively, these local synergy subspaces provide a piecewise nonlinear representation of the reach-to-grasp trajectory, enabling accurate reconstruction and smooth synthesis of joint motions from low-dimensional latent trajectories.

7.3.4 Phase-Aware Trajectory Reconstruction

After learning the local nonlinear synergy subspaces, the complete reach-to-grasp trajectory is reconstructed in joint space by probabilistically blending the local reconstructions according to the HMM state posteriors. Specifically, given an observed joint trajectory $y_{1:T}$, the posterior probability of each latent state is computed as

$$\gamma_{t,n} = P(s_t = n \mid y_{1:T}), \quad t = 1,\ldots,T,\ n = 1,\ldots,N, \tag{7.6}$$

with $\sum_{n=1}^{N} \gamma_{t,n} = 1$ for all t. For each latent state n, a back-constrained GP-LVM has been trained on the corresponding local sub-dataset. By passing the entire sequence $\{y_t\}_{t=1}^{T}$ through the encoder-decoder mapping of the n-th GP-LVM, a reconstructed joint-space trajectory $\{\hat{y}_{t,n}\}_{t=1}^{T}$ is obtained. This yields N locally accurate reconstructed trajectories, denoted Traj_1,…, Traj_N. Each Traj_n achieves its highest reconstruction accuracy in the

time intervals where latent state n is dominant, since those trajectory segments were used to train the corresponding local GP-LVM.

Direct use of the raw posteriors $\gamma_{t,n}$ as blending weights can introduce small but undesirable high-frequency fluctuations near phase boundaries, leading to visible jitter in the reconstructed motion. To mitigate this effect, temporal Gaussian smoothing is applied first to the posterior probabilities. Let $W_{t,n} = \gamma_{t,n}$ denote the initial weight matrix, and let $g_\sigma[k]$ be a discrete Gaussian kernel with standard deviation σ, normalized such that $\sum_k g_\sigma[k] = 1$. The smoothed weight matrix $\tilde{W}$ is obtained by convolving each column along the time axis with nearest-neighbor padding at the boundaries:

$$\tilde{W}_{t,n} = \sum_{k=-r}^{r} g_\sigma[k]\, \hat{W}_{t-k,n}, \tag{7.7}$$

where r is the kernel radius and $\hat{W}_{t,n}$ replicates the boundary values for $t < 1$ and $t > T$. The smoothed weights are then renormalized at each time step to restore their probabilistic interpretation:

$$w_{t,n} = \frac{\tilde{W}_{t,n}}{\sum_{j=1}^{N} \tilde{W}_{t,j} + \varepsilon}, \tag{7.8}$$

with a small ε added for numerical stability.

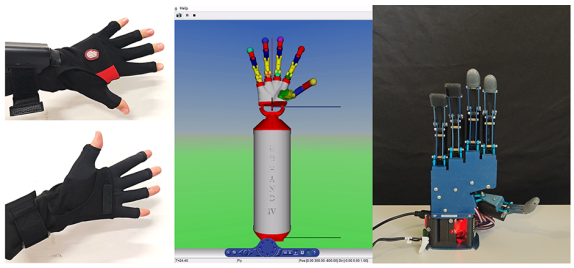
Given the N locally reconstructed trajectories $\hat{y}_{t,n}$ and the corresponding smoothed weights $w_{t,n}$, the final reach-to-grasp trajectory is obtained through probabilistic blending:

$$\hat{y}_t = \sum_{n=1}^{N} w_{t,n}\, \hat{y}_{t,n}, \quad t = 1, \ldots, T. \tag{7.9}$$

This blending strategy produces a continuous and smoothly varying joint trajectory that transitions gradually between locally accurate reconstructions rather than switching abruptly at discrete boundaries. As a result, the reconstructed motion remains physically plausible and temporally consistent throughout the reach-to-grasp process, from the initial open-hand posture to the final stable grasp.

7.4 Quantitative Evaluation and Comparative Analysis

The experimental evaluation assesses the effectiveness of the proposed local probabilistic nonlinear latent representation framework for modeling and reconstructing reach-to-grasp trajectories. Specifically, the experiments examine the impact of local nonlinear synergy



(a) (b) (c)

Figure 7.3 Data collection setup: (a) dataglove, (b) Simulink model of UB Hand IV, (c) AR10 robotic hand.

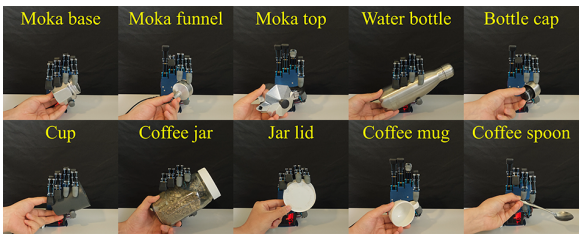


Figure 7.4 Coffee-preparation task dataset with the AR10 robotic hand: ten functional grasps.

representations on final grasp pose reconstruction accuracy, the ability of probabilistic blending to achieve accurate and smooth reconstruction of complete reach-to-grasp trajectories, and the generalization capability of the framework across different robotic hands and grasp scenarios.

To this end, experiments are conducted on two datasets involving distinct robotic hands and task settings. Both quantitative and qualitative evaluations are performed for final grasp pose reconstruction and complete reach-to-grasp trajectory reconstruction, and comparisons are made against linear and nonlinear baseline methods with and without probabilistic segmentation.

7.4.1 Reach-to-Grasp Datasets

Two grasp datasets are used to evaluate the proposed framework and to test its applicability across different robotic hands and task settings. Both datasets are collected using the same teleoperation setup, where a human operator wears a dataglove and performs reach-to-grasp motions while joint angles are streamed in real time to either a simulated or a real robotic hand.

UB Hand IV dataset for typical daily-life grasps

The first dataset is based on a Simulink model of the 15-DoF UB Hand IV, as shown in Fig. 7.3(b). Each trial starts from a fixed open-hand posture and executes one of 31 grasp types defined in (Ficuciello et al. (2014)), and the same grasp-type indexing is adopted in this chapter. For each grasp type, five independent trials are recorded, resulting in a total of 155 reach-to-grasp trajectories. All joint-angle sequences are post-processed using dynamic time warping (DTW) and resampled to a common duration of 5 s at 100 Hz to ensure temporal alignment across demonstrations. This dataset is primarily used to quantitatively evaluate reconstruction accuracy for both final grasp poses and complete reach-to-grasp trajectories under controlled conditions.

AR10 dataset for coffee-preparation functional grasps

To evaluate the proposed method in a more task-oriented and realistic scenario, a second dataset is collected using a real AR10 robotic hand with 10 actuated joints (two DoFs per finger), as illustrated in Fig. 7.3(c). The dataset is designed around a kitchen-related task, namely coffee preparation with a moka coffee maker, and consists of ten functional grasps required to manipulate typical objects involved in this task, such as a moka funnel, a jar lid, and a coffee spoon, as shown in Fig. 7.4. For each grasp type, five demonstrations are performed starting from a standard open-hand posture and progressing to a task-specific final grasp. As with the UB Hand IV dataset, all joint-angle trajectories are temporally aligned using DTW and resampled to a duration of 5 s at 100 Hz. This dataset is used to assess the robustness and generalization capability of the proposed framework when applied to a real robotic hand and task-driven grasps.

7.4.2 Experimental Setup

In all experiments, the left-right Gaussian HMM is configured with $N = 3$ hidden states. The experimental evaluation is organized in two parts. First, the representational capability of the learned synergy subspaces is assessed by reconstructing final grasp poses on the UB Hand IV dataset, which provides a large set of grasp types for rapid and repeatable validation. Second, the reconstruction quality of complete reach-to-grasp trajectories is evaluated on both the UB Hand IV and AR10 datasets to assess spatio-temporal fidelity and cross-hand generalization.

Final grasp pose reconstruction

Final grasp configurations are critical for successful execution, and therefore reconstruction accuracy at the final pose is evaluated first. The reconstruction error is measured by the root mean square error (RMSE) over all actuated joints. For each grasp type, the reported RMSE is obtained by averaging the errors across its five trials. The following methods are considered:

- **PCA**: a global linear synergy model learned from the full trajectories, without HMM-based segmentation.
- **GP-LVM**: a global nonlinear synergy model learned from the full trajectories, without HMM-based segmentation.
- **PCA_SEG**: segment-wise PCA learned on local sub-datasets; final poses are reconstructed by probabilistic blending using smoothed HMM posteriors.
- **GP-LVM_SEG**: segment-wise back-constrained GP-LVM learned on local sub-datasets; final poses are reconstructed using the same blending strategy.

For all methods, two latent dimensionalities ($q = 2$ and $q = 3$, denoted as 2D and 3D) are evaluated to analyze the trade-off between compactness and reconstruction accuracy.

Complete reach-to-grasp trajectory reconstruction

The reconstruction of complete reach-to-grasp trajectories is then evaluated. For the proposed approach, a back-constrained GP-LVM with $q = 3$ is trained on each HMM-induced local sub-dataset, and each local model decodes a full-length joint trajectory. The final reconstructed motion is obtained by probabilistically blending these local reconstructions using temporally smoothed HMM state posteriors, as detailed in Section 7.3.4. Trajectory-level reconstruction accuracy is quantified by computing, at each time step, the RMSE over all joints between the original and reconstructed configurations. A segment-wise PCA baseline with the same latent dimensionality ($q = 3$) and the same HMM-based blending strategy is used for comparison. Experiments are conducted on both datasets to assess reconstruction accuracy and robustness across different hands and grasp sets.

7.4.3 Reconstruction of Final Grasp Poses

Fig. 7.5 summarizes reconstruction errors obtained with different synergy representations. Overall, nonlinear GP-LVM models consistently outperform linear PCA, confirming the

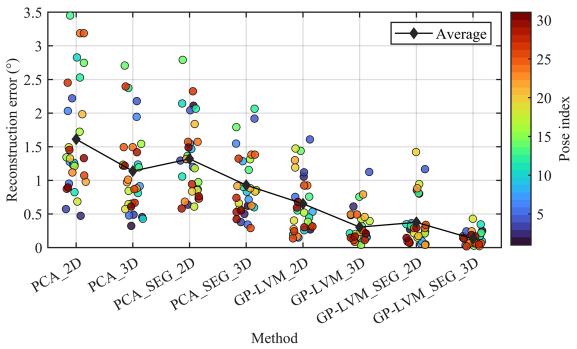


Figure 7.5 Reconstruction errors for different synergy representation methods. Each colored dot corresponds to one grasp type (color-coded by pose index); the black line indicates the mean error for each method.

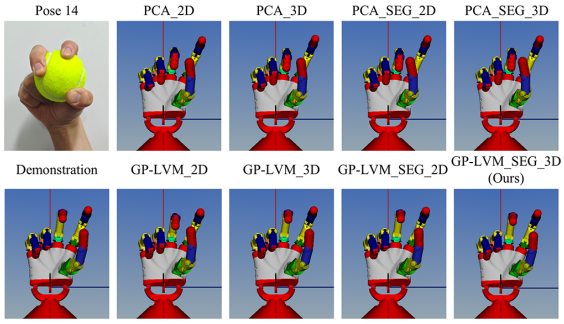


Figure 7.6 Qualitative comparison of reconstructed final grasp poses for pose 14 using different synergy representations.

advantage of nonlinear representations for capturing grasp manifolds. Increasing the latent dimensionality from 2D to 3D reduces reconstruction error for all methods, with a substantially larger improvement observed for GP-LVM. This behavior indicates that correlations among joint variables are strongly nonlinear and cannot be adequately captured by linear projections alone. More importantly, introducing HMM-driven local segmentation further improves reconstruction accuracy for both PCA- and GP-LVM-based representations. By learning separate local synergy subspaces for different portions of the reach-to-grasp motion, the proposed approach reduces interference between heterogeneous regions of the trajectory that would otherwise be forced into a single global latent representation.

Beyond the reduction in mean RMSE, the error distributions obtained with local GP-LVM models are visibly more compact than those of PCA-based methods, indicating increased robustness across different grasps. These results demonstrate that local nonlinear synergy representations provide a more accurate and stable encoding than global linear or nonlinear alternatives.

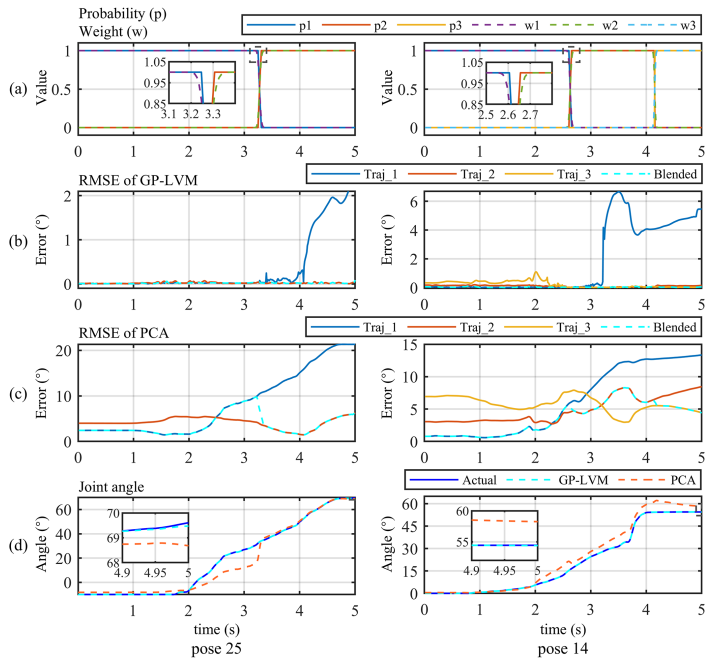


Figure 7.7 Probabilistic blending and reconstruction for two grasps, pose 25 and pose 14. (a) HMM posterior probabilities p_k and smoothed weights w_k, (b) segment-wise and blended reconstruction error for the proposed framework using GP-LVM, (c) error obtained with PCA, (d) example joint-angle trajectories over the full reach-to-grasp sequence.

To further illustrate these differences, Fig. 7.6 provides a qualitative comparison of the reconstructed final grasp poses for a representative example (pose 14). The results show that PCA-based reconstructions exhibit noticeable deviations in finger configuration, particularly in the thumb and distal joints, even when segmentation is applied. In contrast, GP-LVM-based reconstructions more closely match the demonstrated grasp postures, and local modeling further improves the consistency and accuracy of the reconstructed pose. These qualitative observations align with the quantitative trends in Fig. 7.5 and confirm that local nonlinear synergy representations better preserve the geometric structure of complex grasp configurations.

7.4.4 Reconstruction of Complete Reach-to-Grasp Trajectories

Figure 7.7 illustrates the behavior of the proposed framework on two representative UB Hand IV grasps, pose 25 and pose 14. Although the HMM is configured with three latent states, the inferred posterior distributions show that pose 25 is effectively characterized by two dominant local segments, whereas pose 14 exhibits three, indicating that the probabilistic decomposition adapts to the intrinsic structure of each grasp rather than enforcing a rigid

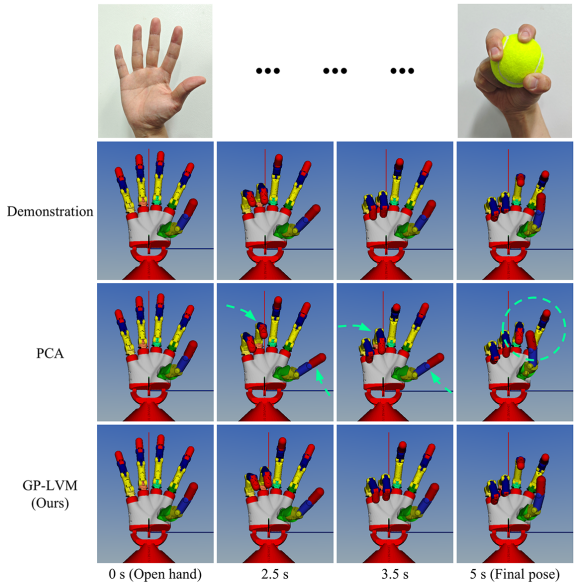


Figure 7.8 Comparison of reconstructed reach-to-grasp motion for pose 14 using PCA and GP-LVM.

partitioning. The HMM state posterior probabilities (p_k) and the corresponding smoothed blending weights (w_k) are shown in Fig. 7.7(a). The zoomed regions highlight the transition intervals, where temporal Gaussian smoothing spreads posterior variations over a short time window. This operation yields smooth cross-fades between local models and avoids abrupt switches that would otherwise introduce kinematic discontinuities in the reconstructed motion.

Figure 7.7(b) and Figure 7.7(c) report the trajectory-level RMSE obtained with local GP-LVM and PCA representations, respectively. For both grasps, each local GP-LVM achieves very low reconstruction error within the time intervals dominated by its associated hidden state, while error increases outside those regions. By probabilistically blending the local reconstructions according to the smoothed HMM posteriors, the final reconstructed trajectory maintains an RMSE that remains close to zero over the entire 5 s reach-to-grasp motion. In contrast, PCA exhibits significantly larger reconstruction errors throughout the trajectory, even when the same HMM-induced local segmentation and blending strategy is applied. This confirms that linear synergies, even when localized, are insufficient to capture the nonlinear spatio-temporal structure of reach-to-grasp motions. Figure 7.7(d) shows an example joint-angle trajectory for each grasp. The GP-LVM-based reconstruction closely follows the measured joint evolution over the entire motion, whereas the PCA-based reconstruction exhibits visible deviations.

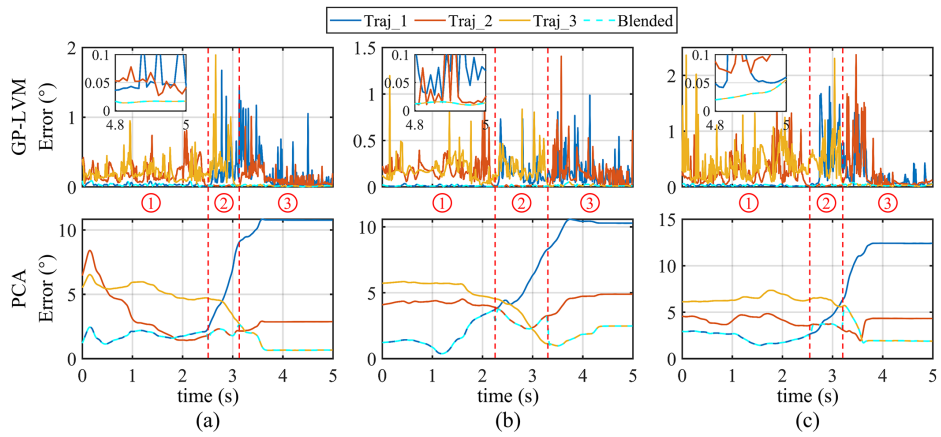


Figure 7.9 RMSE of phase-wise and blended trajectory reconstructions for AR10 coffee-preparation grasps: (a) moka funnel, (b) jar lid, and (c) coffee spoon. The cyan dashed curves indicate the proposed probabilistic blending strategy, yielding low reconstruction error over the full motion; within this framework, GP-LVM achieves lower errors than PCA.

A more intuitive comparison is provided in Figure 7.8, which visualizes the reconstructed reach-to-grasp motion for pose 14. At $t = 2.5$ s and $t = 3.5$ s, the PCA-based reconstruction already shows noticeable discrepancies in the thumb and ring finger, and these deviations become more pronounced at the final grasp. For this type of spherical grasp, such inaccuracies may lead to excessive distributed contact forces, requiring additional force control strategies to ensure safe and stable grasping. In contrast, the GP-LVM-based reconstruction remains closely aligned with the demonstrated hand motion throughout the entire sequence. This visual comparison highlights that local nonlinear synergy representations can faithfully reproduce not only the final posture but also the continuous evolution of the reach-to-grasp, yielding smooth, realistic, and physically consistent trajectories.

The proposed framework is further validated on the real AR10 robotic hand performing coffee-preparation functional grasps. Figure 7.9 shows the trajectory-level RMSE for three representative grasps. The results exhibit the same trend observed in the UB Hand IV experiments: local GP-LVM-based reconstructions consistently outperform PCA-based ones throughout the motion. For each grasp, the local GP-LVM models achieve lower reconstruction errors within the time intervals dominated by their associated hidden states, and probabilistic blending ensures that the final reconstructed trajectory remains close to the minimum achievable error over the full 5 s duration. In contrast, PCA yields substantially higher errors, and even when combined with HMM-based segmentation and blending, the overall reconstruction accuracy remains limited. The final reconstructed poses shown in Figure 7.10 further corroborate these findings. PCA-based reconstructions exhibit visible

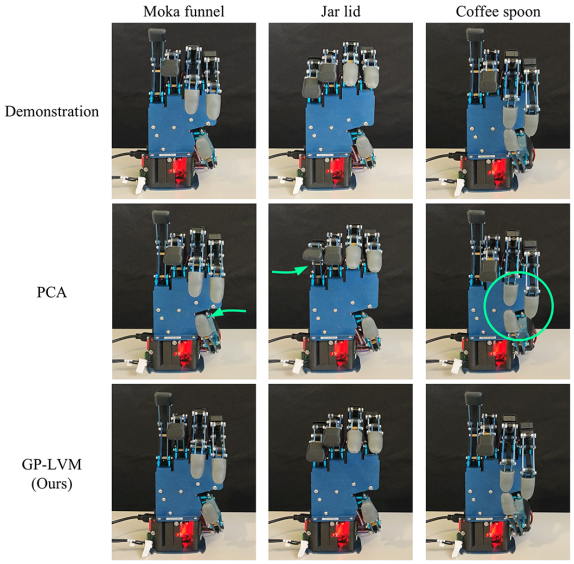


Figure 7.10 Reconstructed final poses of three AR10 coffee-preparation grasps using PCA and GP-LVM.

deviations in finger placement, whereas GP-LVM-based reconstructions closely match the demonstrated AR10 hand configurations. These results indicate that the proposed local probabilistic nonlinear synergy framework remains effective across the two studied robotic hands and grasp sets, providing accurate reach-to-grasp representations in the evaluated synergy subspaces.

7.5 Implications for Phase-Dependent Motion Modeling

This chapter proposes a method based on HMM-driven segmentation and the connection of local nonlinear synergy representations to improve reach-to-grasp trajectory modeling and reconstruction in robotic hands. The approach combines left-right HMMs for probabilistic segmentation, back-constrained GP-LVMs for learning local nonlinear synergy representations, and posterior-based probabilistic blending for reconstructing complete joint-space trajectories. By replacing a single global synergy model with a set of locally valid nonlinear latent representations, the framework captures the heterogeneous and time-varying structure of reach-to-grasp motions more faithfully on the available datasets.

Experimental results on the UB Hand IV and the real AR10 robotic hand show improved reconstruction quality over the considered PCA and global-model baselines for both final grasp poses and complete reach-to-grasp trajectories. The evidence therefore supports local probabilistic nonlinear synergy representations as a compact and effective description for

the studied reach-to-grasp datasets. At the same time, the current datasets remain limited in size and task diversity, so the results should not be interpreted as establishing a universal latent representation for grasping. Within the dissertation, this chapter instead supports the narrower claim that phase-aware probabilistic representations are useful when global latent models become too coarse for time-varying manipulation structure.

Chapter 8

Reproducible Evaluation Infrastructure for Prosthetic Manipulation

8.1 Evaluation Challenges in Prosthetic Manipulation

This chapter complements the method-oriented contributions of the dissertation by addressing uncertainty in evaluation rather than in control or representation alone. Its role is to provide a reproducible infrastructure in which interaction dynamics, sensing, functional metrics, and psychophysical variables can be observed within the same closed loop. The chapter therefore contributes the evaluation pillar of the thesis. Its main limitation is that the current evidence is intentionally proof-of-capability: the reported studies validate platform functionality, not broad clinical efficacy or standardized equivalence to physical assessment.

Over the last decades, upper-limb prostheses have progressed markedly, propelled by improvements in actuators, sensing, and human–machine interfaces that aim to restore manipulation and increase acceptance by users (Farina et al. (2017); Meattini et al. (2018); Silver-Thorn et al. (1996); Tam et al. (2021)). Contemporary myoelectric hands are now designed not only to execute grasps, but also to provide intuitive and stable closed-loop control that supports natural interaction and long-term usability (Clites et al. (2018); Raspopovic et al. (2014)).

Even with these advances, the development and validation of control strategies and feedback schemes remain heavily dependent on hardware experiments, which are expensive, slow, and difficult to reproduce across sites (Atashzar et al. (2021)). Standard functional evaluations usually require access to physical devices, clinical expertise, and controlled settings, which limits their use during early design iterations (Baars et al. (2018)).

Clinical benchmarks such as the Southampton Hand Assessment Procedure (SHAP) are widely used to compare functional performance across grasp types and tasks (Kyberd et al. (2009)). However, SHAP primarily delivers high-level measures like completion time and does not expose internal quantities (joint states, contact forces, or sensorimotor dynamics) that are crucial for control and feedback design (Bruni et al. (2022)).

To address these constraints, simulation has become a key tool for prototyping, validating algorithms, and running preliminary user studies in prosthetics (Sartori et al. (2016)). Physics-based environments such as OpenSim and MuJoCo can model musculoskeletal and robotic systems with sufficient fidelity to study control architectures and interaction dynamics under controlled conditions (Hu et al. (2017); Kumar and Todorov (2015); Todorov et al. (2012)). Simulators also enable the exploration of scenarios that would be impractical or unsafe with physical hardware (Atashzar et al. (2021)).

At the same time, immersive virtual reality (VR) has attracted interest in rehabilitation and assistive robotics because it can increase embodiment, engagement, and sensorimotor learning (Piggott et al. (2016)). Evidence shows that immersive VR supports motor adaptation and skill acquisition in healthy participants and clinical populations, including applications with wearable robotics and neurorehabilitation (Bortone et al. (2020); Rodríguez-Fernández et al. (2024)). These results indicate that VR can offer ecologically valid yet controlled settings for studying human–machine interaction (Piggott et al. (2016)).

Despite this momentum, several limitations persist in current prosthetic simulation platforms. Many rely on generic hands or simplified biomechanics that omit device-specific kinematics, dynamics, and actuation details, reducing the predictive value of the simulations (Bruni et al. (2022)). Immersive interaction is frequently minimized, with third-person or non-immersive interfaces that weaken embodied control (Piggott et al. (2016)). Additionally, functional evaluations implemented in simulation are often ad hoc, which undermines reproducibility and cross-study comparisons (Tam et al. (2021)).

The Hannes prosthetic hand, developed by INAIL and the Italian Institute of Technology, is an ideal case study for tackling these issues due to its anthropomorphic kinematics, underactuated adaptive grasping, and modular design for daily activities (Raspopovic et al. (2014)). Prior work on Hannes explored shared-control grasp pre-shaping and detailed multi-body models for grip-force estimation and interaction analysis (Bruni et al. (2022)). Those efforts, however, were primarily offline or non-immersive and did not provide real-time, human-in-the-loop interaction.

Beyond task performance, modern prosthetic evaluation increasingly targets cognitive and perceptual dimensions of embodiment and integration (Bekrater-Bodmann (2020)). A

central construct is the Sense of Agency, defined as "the experience of controlling one's own motor acts and, through them, the course of external events" (Haggard (2017)). Agency experiences can vary with aging, highlighting their multidimensional character (Mariano et al. (2024)). In upper-limb prosthetics, Sense of Agency is a core element of Prosthesis Embodiment, i.e., "the ability to process information through a prosthesis at the sensory, motor and/or affective levels in the same way as the properties of one's own body parts" (Bekrater-Bodmann (2020)).

Agency can be assessed with explicit questionnaires or implicit psychophysical paradigms (Haggard (2017)). A widely used implicit metric is Intentional Binding, the perceived temporal compression between voluntary actions and their sensory consequences; its magnitude has been linked to agency in multiple contexts (Haggard (2017)). For prosthetics, Intentional Binding provides a quantitative and time-resolved proxy of agency during active device use.

Another perceptual factor relevant to prosthesis use is Phantom Limb Sensation, described as the "phenomenal persistence of postural and sensorimotor features of an amputated limb." Variations in this sensation correlate with user satisfaction and embodiment (Bekrater-Bodmann (2020)). Nevertheless, trial-level monitoring of phantom limb sensations is still rare and often limited to lengthy clinical protocols or retrospective reports.

Immersive simulation can jointly address functional and perceptual assessment by enabling precise control of sensorimotor contingencies while preserving embodied interaction (Piggott et al. (2016)). A framework that unifies device-specific physics, immersive VR, real-time control, sensory feedback, and standardized protocols would therefore be a powerful instrument for prosthetics research and development (Atashzar et al. (2021)).

This chapter introduces iHandVR, an immersive platform for functional testing and evaluation of the Hannes prosthetic hand. The system combines a MuJoCo-based dynamic model with a first-person VR interface built in Unity (Hu et al. (2024)) and a modular ROS-based communication backbone (Quigley et al. (2009)), enabling real-time myoelectric control and vibrotactile feedback. A virtual implementation of SHAP is also included to support standardized, repeatable functional testing.

The studies reported here are intended as proof-of-capability demonstrations rather than hypothesis-driven experiments. A representative SHAP case study and an Intentional Binding protocol are used to validate the platform role itself: the simulator supports both functional and psychophysical evaluations while exposing the kinematic, kinetic, control, and perceptual variables needed for future large-scale investigations. The feasibility of synchronized multimodal data capture within a single immersive platform is therefore established, enabling later statistically grounded studies with larger cohorts.

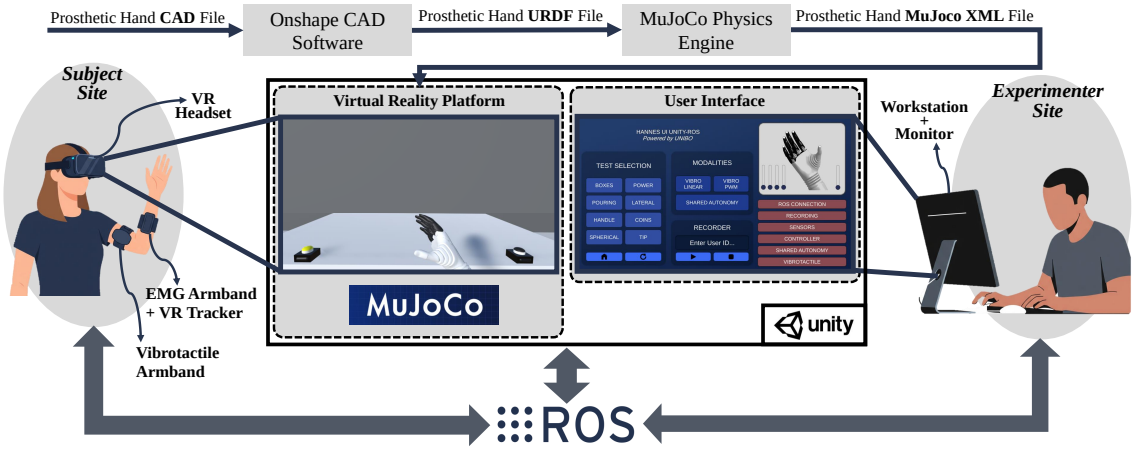


Figure 8.1 System overview with two layers. The upper layer shows the hand-modeling workflow, from CAD to Onshape/URDF/MuJoCo and the resulting XML hand model. The lower layer depicts the VR platform and interaction stack, including Unity visualization, ROS integration, sensor inputs (EMG armband, VR tracker), and outputs (vibrotactile armband).

The main contributions are a device-specific physics model of the Hannes prosthetic hand suitable for real-time interaction; a unified and modular architecture integrating immersive VR, myoelectric control, and vibrotactile feedback; a virtual implementation of standardized functional assessment protocols for prosthetic evaluation; and a demonstration that the platform supports real-time psychophysical paradigms and multimodal data acquisition. Methodologically, the chapter contributes an evaluation infrastructure for closed-loop assessment, in which dynamics, user interaction, and protocol-level measurement are co-designed instead of being treated as separate validation stages.

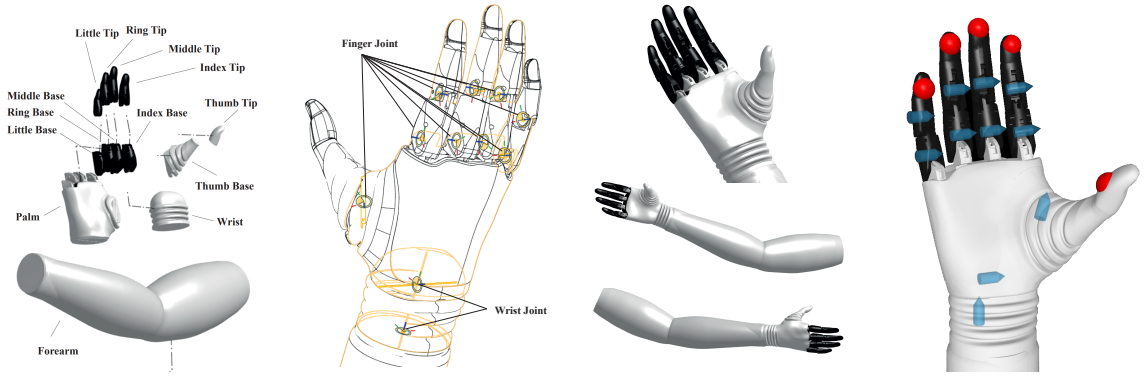
8.2 Integrated Simulation and Interaction Framework

8.2.1 Hannes Hand Modeling Pipeline

The pipeline starts from a 3D mechanical model of the Hannes hand (Fig. 8.2a) and refines it in Onshape[1], a cloud CAD platform, to prepare the geometry for physics-based simulation and VR interaction. Onshape served as the single source of truth for components, joints, and assemblies across design iterations.

Major structural elements, joints, and mounts were modeled and assembled using constraint-based tools. Each finger includes two revolute joints to reproduce proximal

[1]https://www.onshape.com/



(a) (b) (c) (d)

Figure 8.2 Hannes hand model and MuJoCo setup. (a) Exploded view of STL components. (b) Joint placement along fingers, thumb, and wrist. (c) Assembled CAD model. (d) Joint configuration and sensor placement in the MuJoCo simulation model.

and distal flexion, while the thumb uses a single flexion joint for its dominant motion. Two wrist DoFs were defined to represent pronation/supination and flexion/extension (Fig. 8.2b).

To connect CAD with simulation, the assembled model (Fig. 8.2c) was exported via the open-source `onshape-to-robot` tool[2], which produces URDF files. The URDF encodes kinematic hierarchy, joint types and axes, inertial properties, and mesh geometry for visualization and collision.

Although the CAD model provides kinematic structure, joint limits must be explicitly set for dynamic simulation. The URDF joint ranges were iteratively tuned to permit full opening/closing and to cover the grasp postures required by the evaluation tasks.

The URDF model was then converted to MuJoCo-compatible XML. This conversion maps joint definitions, mesh references, and dynamics parameters such as stiffness, damping, and control modes. Contact behavior was configured with MuJoCo parameters (e.g., solref, solimp, friction coefficients), tuned to yield stable grasps and held constant across experiments (Fig. 8.2d).

8.2.2 Kinematic and Dynamic Modeling of the Hannes Hand

The hand model includes 11 joints with different angular limits and viscous damping values, summarized in Table 8.1.

The damping coefficients b in Table 8.1 are implemented in MuJoCo as viscous joint terms and appear in the dynamics as a diagonal matrix $B = \mathrm{diag}(b_0, \ldots, b_{10})$.

[2]https://github.com/Rhoban/onshape-to-robot

Table 8.1 Joint limits, damping values, and symbols.

Joint Name	$[\theta_i^{\min}, \theta_i^{\max}]$ [rad]	$b_i \left[\frac{\mathrm{N\,m\,s}}{\mathrm{rad}}\right]$	Symbol
Wrist Pron/Sup	$[-1.0,\ 1.0]$	0.5	θ_0
Wrist Ext/Flex	$[-1.0,\ 1.0]$	0.5	θ_1
Thumb	$[-2.0,\ 2.0]$	0.05	θ_2
Index Knuckle	$[-0.3,\ 1.5]$	0.05	θ_3
Index Distal	$[0.2,\ 1.7]$	0.05	θ_4
Middle Knuckle	$[-0.3,\ 1.5]$	0.05	θ_5
Middle Distal	$[0.2,\ 1.7]$	0.05	θ_6
Ring Knuckle	$[-0.3,\ 1.5]$	0.05	θ_7
Ring Distal	$[0.2,\ 1.7]$	0.05	θ_8
Little Knuckle	$[-0.3,\ 1.5]$	0.05	θ_9
Little Distal	$[0.2,\ 1.7]$	0.05	θ_{10}

Each joint angle θ_i is defined on its own interval and is mapped to a unitless variable $q_i \in [0,1]$ by an affine transformation, enabling uniform control and algorithmic processing across joints.

Formally, let

$$\theta = \begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_{10} \end{bmatrix}^T, \quad \theta_i \in [\theta_i^{\min}, \theta_i^{\max}], \tag{8.1}$$

be the joint-angle vector with lower and upper bounds $\theta_i^{\min}$ and $\theta_i^{\max}$. The normalized vector $q \in [0,1]^{11}$ is defined component-wise as

$$q_i = \frac{\theta_i - \theta_i^{\min}}{\theta_i^{\max} - \theta_i^{\min}}, \quad i = 0, \ldots, 10, \tag{8.2}$$

where $\Delta\theta_i := \theta_i^{\max} - \theta_i^{\min}$ denotes the range of joint i.

In compact vector form,

$$q = D^{-1}(\theta - \theta^{\min}), \tag{8.3}$$

with

$$\theta^{\min} = \begin{bmatrix} \theta_0^{\min} & \cdots & \theta_{10}^{\min} \end{bmatrix}^T, \quad D = \mathrm{diag}(\Delta\theta_0, \ldots, \Delta\theta_{10}). \tag{8.4}$$

The inverse map from normalized space to joint angles is

$$\theta = Dq + \theta^{\min}. \tag{8.5}$$

This normalization provides a consistent parameterization despite differing mechanical limits and improves the robustness of control, optimization, and learning routines that operate on joint states.

Desired joint angles $\hat{\theta}$ are recovered from normalized references $\hat{q}$ using the inverse affine map in equation (8.5). In MuJoCo, a *position actuator*[3] takes the desired position $\hat{\theta}$ as input for the corresponding joints, and the physics engine applies a torque proportional to the tracking error:

$$\tau = K_p(\hat{\theta} - \theta), \tag{8.6}$$

where $\tau \in \mathbb{R}^{11}$ is the joint-torque vector, $K_p \in \mathbb{R}^{11\times 11}$ is a diagonal matrix of proportional gains, and $\theta \in \mathbb{R}^{11}$ is the current joint-angle vector.

The resulting joint dynamics are second order:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + B\dot{\theta} + g(\theta) = \tau, \tag{8.7}$$

where M, C, and g are the inertia matrix, Coriolis/centrifugal terms, and gravity terms, respectively; B is the damping matrix from Table 8.1; and the applied torque τ is given by the proportional law in equation (8.6).

The convergence rate of θ to $\hat{\theta}$ is set by the gain matrix. In this chapter, the following gain matrix is used:

$$K_p = \mathrm{diag}(\underbrace{100,100}_{\text{joints 0,1}},\underbrace{4,4,4,4,4,4,4,4,4}_{\text{joints 2 to 10}}). \tag{8.8}$$

8.2.3 Sensing and Force Modeling

In the simulator, each fingertip of the Hannes hand is fitted with a tri-axial force sensor measuring contact forces along the local x, y, and z directions. The resulting signals are used to detect contact, assess grasp stability, and drive control or analysis both online and offline.

Let us denote the force vector measured at the fingertip of finger j as

$$f_j = \begin{bmatrix} f_{xj} & f_{yj} & f_{zj} \end{bmatrix}^T \in \mathbb{R}^3, \quad j \in \{T,I,M,R,L\}, \tag{8.9}$$

where T,I,M,R,L denote thumb, index, middle, ring, and little fingers, and f_{xj}, f_{yj}, f_{zj} are the force components in the local frame of fingertip j.

[3] https://mujoco.readthedocs.io/en/stable/XMLreference.html#actuator-position

All fingertip forces can be stacked into a single global vector:

$$F = \begin{bmatrix} f_T^T & f_I^T & f_M^T & f_R^T & f_L^T \end{bmatrix}^T \in \mathbb{R}^{15}, \tag{8.10}$$

so that

$$F = \begin{bmatrix} f_{xT} & f_{yT} & f_{zT} & \cdots & f_{xL} & f_{yL} & f_{zL} \end{bmatrix}^T. \tag{8.11}$$

Each f_j may be used in the local fingertip frame or transformed into a global frame via the known fingertip pose when required for external-force or object-interaction analyses.

The contact-force magnitude at each fingertip is computed with the Euclidean norm:

$$\|f_j\| = \sqrt{f_{xj}^2 + f_{yj}^2 + f_{zj}^2}, \quad \text{for } j \in \{T, I, M, R, L\}. \tag{8.12}$$

In practice, tri-axial measurements are reduced to scalar quantities that are more suitable for system-level outputs and haptic feedback. This compact representation simplifies integration into control loops, vibrotactile feedback computation (Sec. 8.3.3), and learning-based modules (Sec. 8.3.4).

8.2.4 Virtual Reality Platform and Hand Control Interfaces

The simulator offers a first-person, stereoscopic view of the prosthetic workspace. Control is provided through a compact set of normalized input channels that map user intent to grasp and wrist motions, ensuring consistency across tasks and joint ranges.

Unity 3D is used for rendering and interaction, while input/output devices (gaze tracking, controllers, external sensors) are accessed through standardized communication interfaces.

The control interface relies on three scalar channels: $u_0 \in [-1, 1]$ for hand opening ($u_0 < 0$) and closing ($u_0 > 0$); $u_1 \in [-1, 1]$ for wrist extension ($u_1 < 0$) and flexion ($u_1 > 0$); and $u_2 \in [-1, 1]$ for wrist pronation ($u_2 < 0$) and supination ($u_2 > 0$).

Two control modes are implemented: velocity control and position control.

Velocity control mode Each joint $k = 0, \ldots, 10$ has a desired normalized position $\hat{q}_k(t) \in [0, 1]$, with $\hat{q}_k(t) = 0$ representing the fully open/minimum position and $\hat{q}_k(t) = 1$ the fully closed/maximum position. At each time step Δt, the update rule is:

$$\hat{q}_k(t + \Delta t) = \text{clip}\left(\hat{q}_k(t) + \gamma_k(t) u_j(t) \Delta t, 0, 1\right), \tag{8.13}$$

where $\text{clip}(x,a,b) = \min(\max(x,a),b)$, $u_j = u_0$ for finger joints ($k = 2,\ldots,10$), $u_j = u_1$ for wrist extension/flexion ($k = 1$), $u_j = u_2$ for wrist pronation/supination ($k = 0$), and γ_k is a scalar gain that sets the responsiveness of joint k (default $\gamma_k = 1$).

A single closure command $u_0(t)$ thus drives all finger joints, enforcing a consistent grasping synergy. The gain $\gamma_k(t)$ can be tuned online to modulate responsiveness.

Position control mode Alternatively, the same inputs can be used as direct references for normalized joint positions. Because $u_j(t) \in [-1,1]$ must map to $\hat{q}_k(t) \in [0,1]$, the following affine map is used:

$$\hat{q}_k(t) = \begin{cases} \dfrac{u_0(t)+1}{2}, & \text{for } k = 2,\ldots,10, \\ \dfrac{u_1(t)+1}{2}, & \text{for } k = 1, \\ \dfrac{u_2(t)+1}{2}, & \text{for } k = 0. \end{cases} \tag{8.14}$$

Desired joint angles are recovered from normalized positions using the affine mapping in equation (8.3).

Table 8.2 Overview of input and output signals

Type of signal	**Value Range**	**Symbol**
Inputs		
Hand Closure / Opening	$[-1,\ 1]$	u_0
Wrist Flexion / Extension	$[-1,\ 1]$	u_1
Wrist Pronation / Supination	$[-1,\ 1]$	u_2
Arm Pose	$\text{SE}(3)$	$T^{\mathscr{W}}_{\text{tracker}}$
Joint Gain	$\mathbb{R}_+$	γ_k
Outputs		
Fingertip Contact Force	$\mathbb{R}^3$	f_j
Vibrational Feedback Command	$[0,\ 1]$	$d(t)$

8.2.5 Spatial Tracking

The head-mounted display (HMD) provides six degrees of freedom (6DoF) tracking for head and controllers, preserving spatial alignment between the user and the virtual scene. The poses of the HMD and external tracker are represented as homogeneous transforms $T^{\mathscr{W}}_{\text{HMD}}$ and $T^{\mathscr{W}}_{\text{tracker}}$, elements of $\text{SE}(3)$, the group of 3D rigid-body transformations. The prosthetic

forearm is rigidly attached to the tracker with a MuJoCo *weld constraint*[4], enforcing:

$$T_{\text{forearm}}^{\mathscr{W}} = T_{\text{tracker}}^{\mathscr{W}} T_{\text{offset}}, \tag{8.15}$$

where T_{offset} is a fixed transform that aligns the prosthesis with the tracked forearm. This decouples the visual viewpoint from the control frame while keeping hand placement consistent.

8.2.6 System Communication and Software Architecture

The simulation framework uses a modular ROS-based architecture (Melodic Morenia)[5], which manages sensor acquisition, command streams, and logging across modules. The ROS stack runs inside an Ubuntu 18.04[6] Docker container[7] to ensure reproducible deployments.

Unity 2021.3.19f1[8] handles rendering and UI logic and connects to ROS via the ROS-TCP-Connector plugin[9] for bidirectional messaging. The link between Unity and the MuJoCo physics engine (v2.3.7)[10] is provided by a custom integration layer[11] based on the mujoco-unity plugin[12], which keeps state updates synchronized in real time. MuJoCo computes dynamics and contact interactions.

Control and feedback devices (e.g., myoelectric armbands, trackers, vibrotactile actuators) connect via ROS topics and services, enabling plug-and-play integration and closed-loop feedback without modifying the core simulator.

A Unity-based UI supports task selection, system monitoring, and feedback configuration, and can be customized per protocol through modular panels.

8.2.7 Reproducibility Conditions and Validation Metrics

Reproducibility is enforced through explicit control of software, physics, protocol timing, and logging. Software reproducibility is obtained by containerizing the ROS stack and fixing middleware/runtime versions. Physics reproducibility is obtained by keeping MuJoCo model

[4] https://mujoco.readthedocs.io/en/stable/XMLreference.html#equality-weld
[5] https://wiki.ros.org/melodic
[6] https://releases.ubuntu.com/18.04/
[7] https://www.docker.com/
[8] https://unity.com/releases/editor/whats-new/2021.3.19
[9] https://github.com/Unity-Technologies/ROS-TCP-Connector
[10] https://github.com/google-deepmind/mujoco/releases/tag/2.3.7
[11] https://mujoco.readthedocs.io/en/stable/unity.html
[12] https://github.com/deepmind/mujoco-unity

parameters, joint limits, and contact coefficients fixed across runs unless explicitly changed for an experiment. Protocol reproducibility is obtained by fixed trial structures, predefined delay sets, deterministic task reset procedures, and standardized UI-driven phase progression for both IB and SHAP tasks. Measurement reproducibility is obtained by synchronized timestamped logging of control commands, forces, gaze states, task events, and subjective responses within a single ROS time base. Together, these constraints make cross-trial and cross-subject comparisons technically auditable.

8.3 Experimental Protocols and Multimodal Synchronization

This section outlines the experimental setup, adaptive myoelectric control, and feedback strategies used to evaluate the VR simulator. Two protocols are then presented: Intentional Binding as an agency-related demonstration (Sec. 8.3.5) and a virtual SHAP for functional assessment (Sec. 8.3.6).

8.3.1 Experimental Setup

The experimental setup was designed to assess how well the VR simulator supports advanced human–prosthesis interaction strategies.

Hardware Components

The simulation framework relies on hardware chosen for high performance, immersive interaction, and real-time responsiveness:

- **Workstation:** A Windows 11 desktop equipped with an Intel Core i7-14700F CPU, 32 GB DDR4 RAM, and an NVIDIA GeForce RTX 4060 GPU.
- **VR Headset:** The *HTC VIVE Pro Eye* provides 1440×1600 pixels per eye, a 90 Hz refresh rate, and full 6-DoF tracking.
- **sEMG Acquisition System:** Two *Myo Armbands* (Thalmic Labs) worn on the user's forearm (or residual limb) record muscle activity and provide eight-channel surface EMG data, nine-axis IMU data, and wireless transmission via Bluetooth. A short calibration recording is collected for seven postures (rest, open, power, flexion, extension,

pronation, and supination). EMG signals are streamed in real time and processed by ROS nodes for intent estimation and prosthesis control.

8.3.2 Incremental Learning for Myoelectric Control

To maintain robust control under non-stationary electromyographic (EMG) signals, we use an Incremental Learning (IL) strategy that continuously updates the myoelectric model as new data arrive (Braun et al. (2025)). This enables real-time adaptation to changing muscle activation patterns without full retraining or loss of prior knowledge.

Let $\mathscr{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ be the dataset collected at time step t, where $x_i \in \mathbb{R}^m$ is the EMG feature vector for sample i with m features extracted from raw sEMG (e.g., RMS, mean absolute value, waveform length), $y_i \in \mathbb{R}^3$ is the desired output vector of normalized commands for the three prosthetic DoFs (hand closure u_0, wrist flexion u_1, wrist pronation u_2), and N_t is the number of samples at time t.

The control model is a parametric mapping:

$$f_{\beta_t} : \mathbb{R}^m \to [-1,1]^3,$$

where $\beta_t \in \mathbb{R}^p$ denotes the model parameter vector at time t, and p is the number of trainable parameters.

Given an input vector $x(t) \in \mathbb{R}^m$, the model outputs the control command:

$$u(t) = f_{\beta_t}(x(t)) = \begin{bmatrix} u_0(t) \\ u_1(t) \\ u_2(t) \end{bmatrix} \in [-1,1]^3, \tag{8.16}$$

where each $u_j(t)$ is the normalized control signal for DoF j ($j = 0,1,2$).

Incremental updates are performed with a single gradient descent step:

$$\beta_t = \beta_{t-1} - \eta \, \nabla_\beta \mathscr{L}(\beta_{t-1}; \mathscr{D}_t), \tag{8.17}$$

where $\eta > 0$ is the learning rate and $\mathscr{L}(\cdot)$ is the loss:

$$\mathscr{L}(\beta; \mathscr{D}_t) = \frac{1}{N_t} \sum_{i=1}^{N_t} \left\| f_\beta(x_i) - y_i \right\|^2 + \lambda \, R(\beta, \beta_{t-1}). \tag{8.18}$$

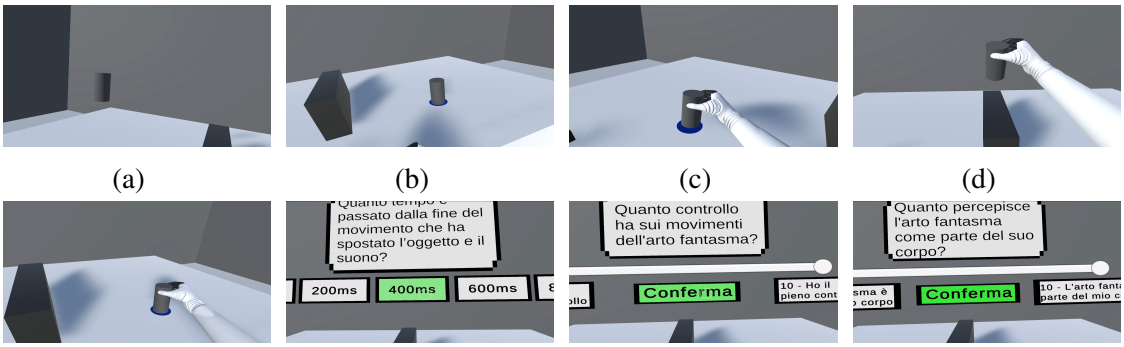


Figure 8.3 Intentional Binding experiment in the VR environment. (a–b) Passive phase: the cylinder moves from the initial to the target platform without user control. (c–e) Active phase: the user grasps, lifts, transports, and releases the object via myoelectric control of the virtual prosthetic hand. (f–h) Response phase: gaze-driven UI presenting the *Perceived Delay*, *Control over Phantom Limb*, and *Phantom Limb Embodiment* questions: *"How much time elapsed between the end of the movement that displaced the object and the sound?"*, *"How much control did you feel you had over the movements of the phantom limb?"*, and *"To what extent did you perceive the phantom limb as part of your body?"*, including selection and confirmation.

The first term is the mean squared error over the current batch, and the second term is a regularization penalty, where $\lambda \geq 0$ sets the regularization strength and $R(\beta, \beta_{t-1})$ is a proximity term (e.g., $R = \|\beta - \beta_{t-1}\|^2$) that discourages abrupt parameter changes between updates.

This formulation lets the model adapt smoothly to temporal changes in EMG features while maintaining stable control. The learned mapping f_{β_t} therefore provides intuitive, simultaneous real-time control of the three DoFs.

8.3.3 Vibrotactile Feedback Strategies

The vibrotactile system delivers stimulation through a custom wristband with a vibration motor. It is controlled wirelessly via Bluetooth and modulates vibration intensity based on simulated contact forces, closing the sensorimotor loop and supporting embodiment.

Two feedback strategies are provided. A saturation threshold $F_{\max}$ is defined; here $F_{\max} = 10\text{N}$ is used as a reasonable interaction-force limit. In both modes, a normalized force level $r(t) \in [0,1]$ is computed from the thumb force:

$$r(t) = \min\left(\max\left(\frac{\|f_T(t)\|}{F_{\max}}, 0\right), 1\right). \tag{8.19}$$

Figure 8.4 Researcher-side UI for the Intentional Binding protocol, with experiment-order controls, delay counters, reset actions, and system status/recording panels including hand visualization.

Proportional Vibration Mode In this mode, vibration intensity changes continuously with contact force, so the output command is:

$$d(t) = r(t). \tag{8.20}$$

The resulting vibration intensity scales linearly with the measured force.

Pulse Width Modulation (PWM) Vibration Mode In this mode, the motor is driven by a binary (ON/OFF) square wave with fixed frequency and variable duty cycle.

Define the phase within each PWM period as $\phi(t) = t/T_{\text{PWM}} - \lfloor t/T_{\text{PWM}} \rfloor \in [0,1)$. We set $T_{\text{PWM}} = 0.5\,\text{s}$, which yields a stable and clearly perceivable modulation, while remaining a tunable parameter. The PWM output command $d(t) \in \{0,1\}$ is:

$$d(t) = \begin{cases} 1, & \text{if } \phi(t) < r(t), \\ 0, & \text{otherwise}, \end{cases} \tag{8.21}$$

This produces a pulse waveform whose ON duration is proportional to interaction force, yielding a temporally modulated vibration pattern with fixed amplitude and variable pulse width.

8.3.4 Shared-Autonomy Gain Modulation

As an experiment-level capability, a shared-autonomy layer adapts the control gain γ based on interaction state, inspired by probabilistic shared-autonomy approaches. Let $u(t) = [u_0(t)\, u_1(t)\, u_2(t)]^T \in [-1,1]^3$ denote the incremental-learning command vector (hand closure, wrist flexion, wrist pronation). For hand closure, the desired normalized position is updated

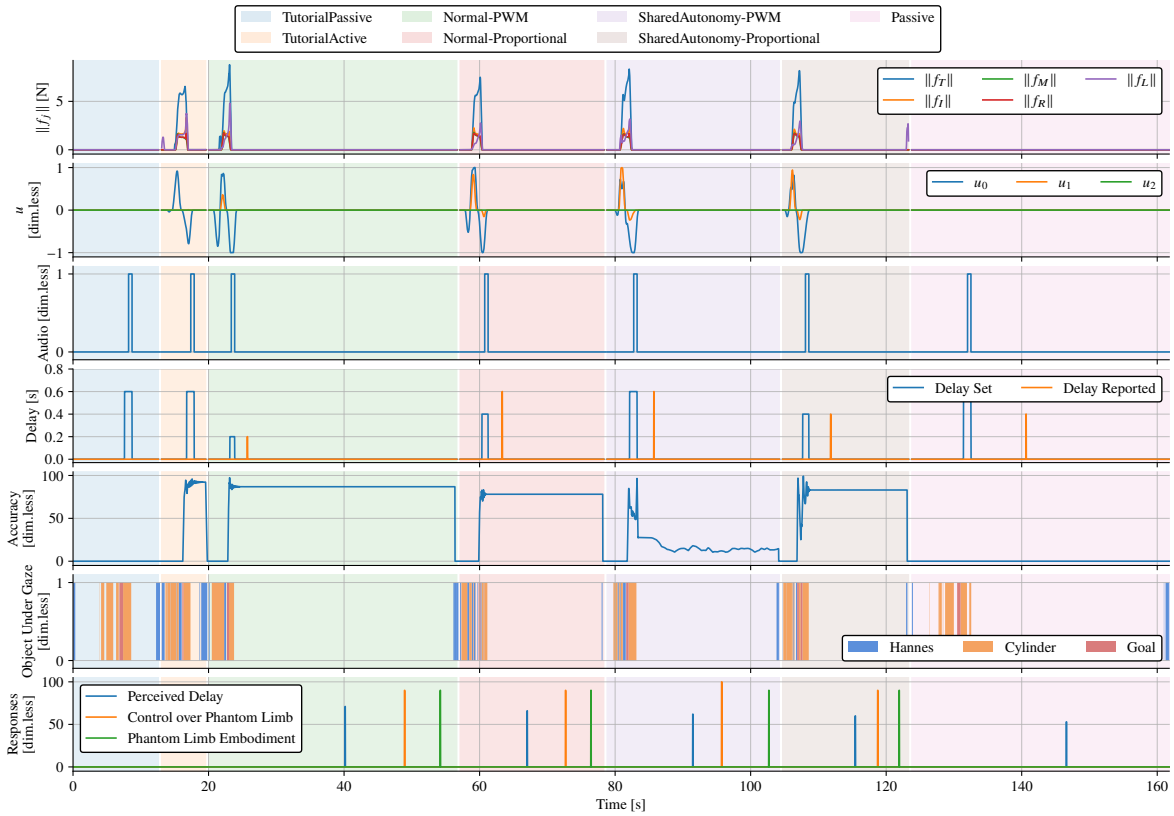


Figure 8.5 Synchronized overview of a representative Intentional Binding session. The plot shows fingertip force magnitudes $\|f_j\|$, control inputs/velocities u, audio trigger, imposed and reported delays, placement accuracy, object-under-gaze state, and the three subjective ratings (Perceived Delay, Control over Phantom Limb, Phantom Limb Embodiment). Shaded bands indicate task phases/conditions. One trial per phase is shown.

via the velocity-control rule in Eq. (8.13), with γ scaling the update after applying $u_0(t)$. Wrist commands $u_1(t)$ and $u_2(t)$ are instead mapped directly to position references as in the position-control mode (Eq. (8.14)). Let $O(t) = [u_0(t) \;\; F(t)]^T$ collect the open/close command and contact force magnitude. A compact state estimator provides online state probabilities $p_j(t)$ for $j = 1,\ldots,N_s$, where N_s is the number of interaction states; free-motion and contact-regulation states are associated with high and low gains, respectively. The gain is computed as a weighted combination, with $\kappa_j \in [\gamma_{\min}, \gamma_{\max}]$ the gain assigned to state j:

$$\gamma(t) = \sum_{j=1}^{N_s} \kappa_j \, p_j(t), \qquad \sum_{j=1}^{N_s} p_j(t) = 1, \tag{8.22}$$

and applied as a shared gain to the hand-closure joints, i.e., $\gamma_k(t) = \gamma(t)$ for all finger joints $k = 2,\ldots,10$, reflecting the single DoF for opening/closing. In our implementation, $\gamma_{\max} = 1$

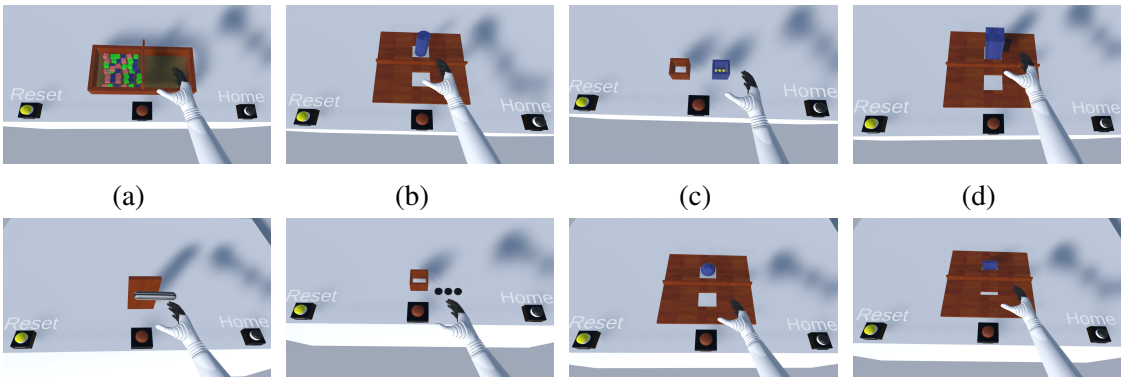


Figure 8.6 Virtual SHAP implementation in the VR environment. (a) Boxes task for grasping and transporting cubes; (b) Power grasp task for gross strength and control; (c) Pouring task to assess coordination during liquid transfer; (d) Lateral grasp task for precision pinch between thumb and lateral index; (e) Handle task emulating manipulation of a handled object; (f) Coins task for fine dexterity with small objects; (g) Spherical grasp task to evaluate palmar grasp of a round object; (h) Tip grasp task emphasizing fingertip control.

and $\gamma_{\min} = 0.1$, while keeping these limits user-configurable. This adaptation maintains responsiveness in free space ($\gamma \approx \gamma_{\max}$) and supports fine force regulation during contact ($\gamma \approx \gamma_{\min}$).

8.3.5 Intentional Binding Experiment

To demonstrate support for psychophysical paradigms tied to agency and sensorimotor integration, we implemented an experiment based on Intentional Binding (IB). In this context, the protocol serves as a feasibility check for closed-loop timing and agency-related interaction rather than a neuroscience study. IB is a temporal effect in which actions and their outcomes are perceived as closer in time and is commonly used as a proxy for agency.

Experimental Protocol

A simple pick-and-place scenario was used in VR. The scene contains a white table, a blue pick marker on the left, a blue place marker on the right, and a rigid black vertical obstacle in the center, as shown in Figure 8.3. With myoelectric control, participants grasped the object using the Hannes hand and moved it to the right platform. After successful placement, a beep was triggered following a fixed delay D_f.

Each trial followed a fixed sequence: the participant grasped and lifted the object from the left platform, moved it over the obstacle and released it on the right platform, a beep

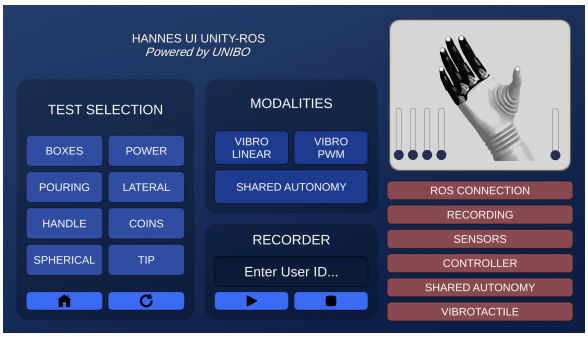


Figure 8.7 Researcher-side UI for the virtual SHAP protocol, including task selection, modality toggles (vibrotactile mode and shared-autonomy mode), recording controls, and system status panels with hand visualization.

was delivered through headphones after the fixed delay D_f, and the participant estimated the perceived delay D_p.

Before data collection, participants completed a brief training phase to familiarize themselves with the timing and feedback. The protocol included seven phases (TutorialPassive, TutorialActive, Normal-PWM, Normal-Proportional, SharedAutonomy-PWM, SharedAutonomy-Proportional, Passive): a passive tutorial in which the cylinder moved autonomously and the imposed delay was shown via a graphical interface; an active familiarization phase in which the participant moved the cylinder while still being trained on the delay and informed of D_f; and five experimental phases. Four of these were active phases combining *Shared-Autonomy* (Section 8.3.4) with a vibration mode (Section 8.3.3), and one was passive; all experimental phases required responses to the full set of questions. The protocol used 25 trials for each tutorial phase and 24 trials for each non-tutorial phase. During trials, D_f was selected at random from a predefined set (1 ms, 200 ms, 400 ms, 600 ms, 800 ms). After each trial, participants entered their estimate D_p using a gaze-driven UI. During this response phase (Figure 8.3), the UI also asked the *Perceived Delay*, *Control over Phantom Limb*, and *Phantom Limb Embodiment* questions: *"How much time elapsed between the end of the movement that displaced the object and the sound?"*, *"How much control did you feel you had over the movements of the phantom limb?"*, and *"To what extent did you perceive the phantom limb as part of your body?"*.

System Integration

The experiment ran entirely inside the iHandVR framework. EMG-based control enabled real-time manipulation of the virtual prosthesis. The system reliably detected object placement and triggered the auditory beep with precisely controlled latency.

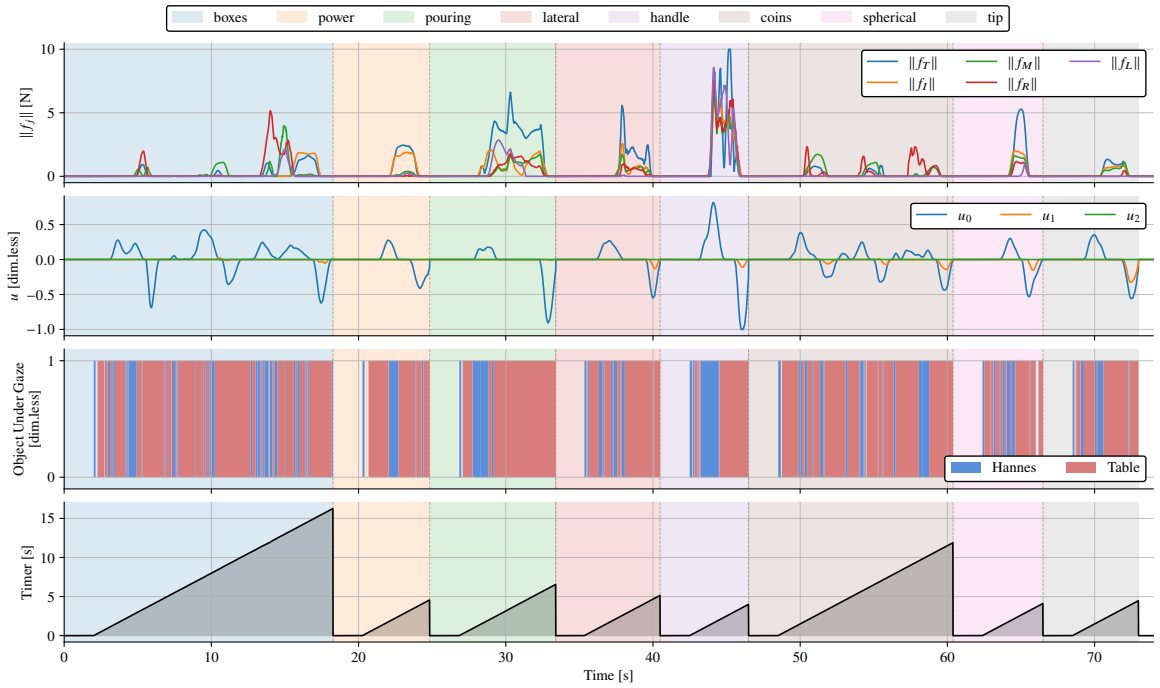


Figure 8.8 SHAP simulator signals for quantitative analysis. The plot reports fingertip force magnitudes $\|f_j\|$, velocity components, gaze state, and the task timer over the selected trial window; task segments are marked by colored spans and transitions by dashed vertical lines. These synchronized signals enable objective verification of timing, motor output, and visual attention, and support reproducible comparisons across trials and subjects.

The researcher-side UI used to set experiment order, delays, and reset actions is shown in Fig. 8.4.

Figure 8.5 presents a synchronized overview of a representative session, including forces, inputs, timing, gaze, and subjective ratings.

The vibrotactile wristband remained connected and switched vibration modes across phases.

Outcome and Significance

Figure 8.3 summarizes the passive, active, and response phases in VR, while Figure 8.5 provides a synchronized example of forces, commands, timing, gaze, and subjective ratings within a trial. Together, these results show that the simulator can execute the full Intentional Binding protocol with controlled timing and synchronized measurements.

The experiment demonstrated that the simulator can integrate real-time action detection with precisely timed feedback, support psychophysical protocols that require subjective responses, and provide immersive interaction suitable for agency-related paradigms.

Although the study was not intended to analyze intentional binding effects themselves, the successful execution confirms that iHandVR can support complex human-in-the-loop evaluations involving cognitive and perceptual variables.

8.3.6 Virtual SHAP Protocol

To demonstrate support for standardized functional testing, we implemented a virtual version of the Southampton Hand Assessment Procedure (SHAP), a widely used protocol for evaluating upper-limb prostheses during task-based manipulation.

Experimental Protocol

The SHAP test was integrated into the VR environment using the v1.31 task suite available from the MuJoCo community for the Modular Prosthetic Limb[13]. The tasks were adapted to the Hannes hand model and embedded in the virtual scene to assess different grasp types under controlled conditions.

The SHAP-inspired tasks included the *Boxes Task* for transporting colored blocks, the *Power Grasp Task* for strength and stability with a handled object, the *Pouring Task* for coordination during liquid transfer, the *Lateral Grasp Task* for lateral pinch between thumb and index finger, the *Handle Task* for door-handle manipulation, the *Coins Task* for fine dexterity with small objects, the *Spherical Grasp Task* for palmar grasp of a virtual ball, and the *Tip Pinch Task* for fingertip placement into a precision slot.

Each task includes a start configuration and a target location. A red button, mirroring the physical SHAP button, starts the timer when pressed and stops it when pressed again. A reset button restarts the current task, while a home button returns to the task-selection menu; once at home, the next task is chosen via a gaze-driven UI. The scene is reset between tasks to preserve repeatability, and object poses are fixed to avoid confounds when comparing grasp types.

The researcher-side control panel used to select tasks and manage recording is shown in Figure 8.7.

For each trial, the system automatically recorded completion times, fingertip forces, user commands, and gaze signals, enabling synchronized analysis of task performance and interaction dynamics.

[13] MuJoCo community resource: Modular Prosthetic Limb SHAP test suites

Outcome and Significance

Figure 8.6 illustrates the SHAP task set implemented in VR, and Figure 8.8 reports synchronized execution time, interaction forces, kinematic variables, and gaze state for a representative trial. The plot shows task segmentation alongside fingertip force norms and gaze state, allowing direct inspection of contact timing and force evolution in each phase. These results indicate that iHandVR can run SHAP-inspired functional tests with consistent virtual structure while enriching evaluation with multimodal data, supporting repeatable, instrumented assessment and motivating future user studies.

8.4 Implications for Standardized Assessment

8.4.1 Functional Scope of the Simulator

Results from the virtual SHAP preserve the intended distinction between precision- and power-grasp tasks and provide synchronized force and gaze profiles across tasks (Figs. 8.6–8.8). The Intentional Binding protocol runs end-to-end with controlled timing and synchronized action, force, gaze, and subjective measures (Figs. 8.3–8.5). Together, these findings indicate that the simulator captures task-level functional demands within the virtual environment, yields richer signals than completion time alone, and supports psychophysical testing within a single platform. They do not, by themselves, establish equivalence to physical SHAP administration or predict real-world prosthetic performance.

8.4.2 Advantages and Current Limitations

The modular architecture simplifies integration of new sensors and feedback devices, enabling rapid comparison of control and feedback strategies under standardized virtual conditions. At the same time, the current evaluation is limited in scope and relies on virtual tasks, and contact/compliance parameters are tuned for qualitative realism rather than hardware calibration. The present results therefore support repeatability and reproducibility orientation within the simulator, but not yet real-world transfer or equivalence to clinical testing. Validation against the physical Hannes hand and further user studies, including prosthesis users, are still needed.

8.4.3 Implications for Prosthetic Research and Development

By providing a repeatable immersive environment, iHandVR supports rapid prototyping of control schemes, training protocols, and feedback strategies before hardware trials. The availability of synchronized interaction signals can also accelerate data-driven development and benchmarking for future prosthetic systems. More importantly, the platform operationalizes evaluation as a closed loop: control design, interaction dynamics, and assessment metrics are observed in the same experimental cycle, which strengthens reproducibility and comparability.

Chapter 9

Conclusions and Future Directions

9.1 Recall of the Problem and Objectives

This dissertation starts from the following scientific question: how can effective robotic manipulation be achieved when uncertainty affects the interaction itself, the sensing process, the control interface, and the evaluation setting? Across industrial manipulation and prosthetic-oriented control, the central challenge is not only generating motion, but doing so effectively when object behavior, contact conditions, and human input are only partially known.

Accordingly, the objective was not to optimize isolated modules, but to integrate sensing, modular human-in-the-loop programming, adaptive control, probabilistic inference, and reproducible evaluation into a single methodological framework for robotic manipulation under interaction uncertainty. With the inclusion of Chapter 4, this framework explicitly covers not only how the robot senses and acts, but also how task knowledge is interactively programmed and updated in collaborative industrial settings.

9.2 Synthesis of the Main Results

This thesis delivers six integrated contributions.

1. An adaptive data-driven optimization framework for elastic manipulation, showing that stiffness-aware evolutionary search can exploit interaction measurements during path generation when elastic uncertainty affects task quality (Chapter 2).

2. A sensing-rich mechatronic architecture for dual-arm wiring harness assembly, demonstrating end-to-end execution of cable routing and connector-related operations with multimodal sensing and modular execution logic (Chapter 3).

3. A modular human-in-the-loop design pattern for industrial robot programming, showing that multimodal interaction, explicit supervision, and incremental retraining can structure task-level programming in collaborative connector assembly; in the reported study, 10 subjects achieved an average full-procedure completion time of 861.7 s and an autonomous insertion success rate of 80.0% after programming, which should be interpreted as proof-of-concept evidence rather than as broad usability validation (Chapter 4).

4. A probabilistic shared-autonomy strategy for grip-force regulation and force-aware skill encoding, where sEMG intent and tactile measurements are integrated through phase inference to improve control smoothness, controllability, and teachability in human-in-the-loop manipulation (Chapters 5 and 6).

5. A local probabilistic nonlinear latent representation for reach-to-grasp reconstruction, improving fidelity over global and linear baselines in phase-dependent motions (Chapter 7).

6. An immersive evaluation platform (iHandVR) integrating real-time control, multimodal sensing, and standardized functional/psychophysical protocols for reproducible prosthetic assessment across virtual interaction scenarios (Chapter 8).

Taken together, these results support one unified conclusion: effective robotic manipulation under interaction uncertainty does not emerge from any single module in isolation, but from the integration of informative sensing, modular task programming, adaptive control, probabilistic inference, and reproducible evaluation across the full manipulation pipeline.

9.3 Critical Evaluation of the Work

The main strength of this dissertation is cross-level coherence: the same interaction-uncertainty perspective is maintained from path generation to system integration, through human-in-the-loop task programming, from online control to latent representation, and from algorithm validation to immersive evaluation. However, the current validity domain has clear limits.

- Several methods rely on quasi-static interaction assumptions to preserve tractability and real-time performance.
- Generalization has been validated on a limited set of tasks, users, and environmental conditions.
- The industrial human-in-the-loop programming contribution is currently demonstrated on a representative connector-assembly workflow rather than across a broad catalog of collaborative tasks.
- Some pipelines require task-dependent calibration and tuning, which can slow transfer to new setups.
- Performance depends on hardware-specific sensing quality and synchronization, especially for tactile and sEMG channels.

Therefore, this dissertation does not claim a complete solution for arbitrary unstructured manipulation. Its claim is narrower and testable: integrating sensing, modular task programming, adaptive control, probabilistic inference, and reproducible evaluation yields measurable gains in effectiveness under structured interaction uncertainty.

9.4 Scientific and Application Implications

The scientific implication is a clearer paradigm for robotic manipulation: uncertain interaction should be treated as a first-class design condition rather than as residual variability around an otherwise deterministic problem.

Methodologically, this means that sensing, task programming, control, inference, and evaluation should be designed as a coupled framework rather than as loosely connected stages. Within this framework, interaction phases become active control variables rather than post-hoc labels, data-driven optimization becomes preferable where analytical interaction models are difficult to identify, modular human-in-the-loop programming becomes essential where tasks must be reconfigured by operators, and probabilistic latent representations become useful when temporal structure cannot be captured by global deterministic models. At application level, this supports more effective industrial handling of elastic components, more structured collaborative robot teaching, and more controllable prosthetic human-in-the-loop operation. It also promotes a stronger evaluation standard in which claims are tied to synchronized multimodal evidence across both physical and virtual settings.

9.5 Future Research Directions

Future work should focus on five high-impact directions.

1. **Learning from interaction-rich data:** integrate learning-based models that autonomously extract manipulation strategies from force, tactile, motion, and intent histories.

2. **Richer sensing and observability:** move toward distributed, higher-bandwidth tactile and multimodal sensing to improve observability of local interaction states.

3. **Scalable human-in-the-loop task programming:** extend the Chapter 4 framework toward richer multi-step assembly procedures, additional multimodal channels, and systematic usability and transferability benchmarks.

4. **Dynamic and non-stationary interaction handling:** extend the framework from quasi-static assumptions to fast, strongly dynamic, and non-stationary interactions.

5. **Personalized human–robot interaction at scale:** develop adaptive co-manipulation and shared-autonomy policies that personalize assistance across users, devices, and task contexts.

These directions maintain the same core thesis while extending its validity domain: integrated manipulation under interaction uncertainty should evolve from structured laboratory settings toward more general, data-efficient, and human-centered robotic systems.

References

Abi-Farraj, F., Pacchierotti, C., Arenz, O., Neumann, G., and Giordano, P. R. (2019). A haptic shared-control architecture for guided multi-target robotic grasping. *IEEE Transactions on Haptics*, 13(2):270–285.

Adjigble, M., Marturi, N., Ortenzi, V., and Stolkin, R. (2019). An assisted telemanipulation approach: combining autonomous grasp planning with haptic cues. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3164–3171. IEEE.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schaffer, A., Kosuge, K., and Khatib, O. (2018). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975.

Antfolk, C., D'alonzo, M., Rosén, B., Lundborg, G., Sebelius, F., and Cipriani, C. (2013). Sensory feedback in upper limb prosthetics. *Expert review of medical devices*, 10(1):45–54.

Asfour, T., Azad, P., Gyarfas, F., and Dillmann, R. (2008). Imitation learning of dual-arm manipulation tasks in humanoid robots. *International Journal of Humanoid Robotics*, 5(02):183–202.

Atashzar, S. F., Carriere, J., and Tavakoli, M. (2021). How can intelligent robots and smart mechatronic modules facilitate remote assessment, assistance, and rehabilitation for isolated adults with neuro-musculoskeletal conditions? *Frontiers in Robotics and AI*, 8:610529.

Baars, E. C., Schrier, E., Dijkstra, P. U., and Geertzen, J. H. (2018). Prosthesis satisfaction in lower limb amputees: a systematic review of associated factors and questionnaires. *Medicine*, 97(39):e12296.

Bdiwi, M., Pfeifer, M., and Sterzing, A. (2017). A survey of human–robot collaboration in industrial settings. *Procedia CIRP*, 63:23–28.

Bekiroglu, Y., Laaksonen, J., Jorgensen, J. A., Kyrki, V., and Kragic, D. (2011). Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616–629.

Bekrater-Bodmann, R. (2020). Perceptual correlates of successful body–prosthesis interaction in lower limb amputees: psychometric characterisation and development of the prosthesis embodiment scale. *Scientific reports*, 10(1):14203.

Bernardin, K., Ogawara, K., Ikeuchi, K., and Dillmann, R. (2003). A hidden markov model based sensor fusion approach for recognizing continuous human grasping sequences. In *Proc. 3rd IEEE International Conference on Humanoid Robots*, pages 1–13.

Bernardin, K., Ogawara, K., Ikeuchi, K., and Dillmann, R. (2005). A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Transactions on Robotics*, 21(1):47–57.

Bernardini, A., Meattini, R., Pasquali, A., Laudante, G., Gentile, C., Gruppioni, E., Palli, G., and Melchiorri, C. (2025). HMM-based shared autonomy for grip strength regulation in sEMG-driven robot hand control. Submitted to IEEE Transactions on Robotics.

Bicchi, A., Gabiccini, M., and Santello, M. (2011). Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581):3153–3161.

Bortone, I., Barsotti, M., Leonardis, D., Crecchi, A., Tozzini, A., Bonfiglio, L., and Frisoli, A. (2020). Immersive virtual environments and wearable haptic devices in rehabilitation of children with neuromotor impairments: a single-blind randomized controlled crossover pilot study. *Journal of neuroengineering and rehabilitation*, 17(1):144.

Boussaïd, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences*, 237:82–117.

Braun, H., Sierotowicz, M., Egle, F., Scheidl, M.-A., Montenegro, S. M., Thuerauf, S., and Castellini, C. (2025). Introducing MOSAIC: A Modular, Open-Source Suite for Assistive Intelligent Control. *IEEE Access*, 13:213713–213733.

Bruni, G., Bucchieri, A., Tessari, F., Boccardo, N., Marinelli, A., De Momi, E., Laffranchi, M., and De Michieli, L. (2022). A multi-body model of an upper-limb prosthesis for grip force estimation and related object interaction application. In *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 01–07. IEEE.

Burdet, E., Franklin, D. W., and Milner, T. E. (2013). *Human robotics: neuromechanics and motor control*. MIT press.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. G. (2010). Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 17(2):44–54.

Canal, G., Alenya, G., and Torras, C. (2022). Capirci: A conversational interface for human–robot collaboration. *IEEE Robotics and Automation Letters*, 7(1):318–325.

Caporali, A., Kicki, P., Galassi, K., Zanella, R., Walas, K., and Palli, G. (2024). Deformable linear objects manipulation with online model parameters estimation. *IEEE Robotics and Automation Letters*, pages 1–8.

Caporali, A., Mirto, M., Pirozzi, S., and Palli, G. (2026). Vision and tactile sensing for DLO manipulation and pin insertion in robotic connector assembly. *IEEE/ASME Transactions on Mechatronics*.

Chelouah, R. and Siarry, P. (2000). A continuous genetic algorithm designed for the global optimization of multimodal functions. *Journal of Heuristics*, 6:191–213.

Childress, D. S. (1980). Closed-loop control in prosthetic systems: historical perspective. *Annals of biomedical engineering*, 8(4):293–303.

Ciocarlie, M. and Allen, P. (2010). Data-driven optimization for underactuated robotic hands. In *ICRA*, pages 1292–1299.

Ciocarlie, M. T. and Allen, P. K. (2009). Hand posture subspaces for dexterous robotic grasping. *IJRR*, 28(7):851–867.

Cipriani, C., Zaccone, F., Micera, S., and Carrozza, M. C. (2008). On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction. *IEEE Transactions on Robotics*, 24(1):170–184.

Cirillo, A., Costanzo, M., Laudante, G., and Pirozzi, S. (2021). Tactile sensors for parallel grippers: Design and characterization. *Sensors*, 21(5):1915.

Clites, T. R., Carty, M. J., Ullauri, J. B., Carney, M. E., Mooney, L. M., Duval, J.-F., Srinivasan, S. S., and Herr, H. M. (2018). Proprioception from a neurally controlled lower-extremity prosthesis. *Science Translational Medicine*, 10(443):eaap8373.

Colledanchise, M. and Ogren, P. (2021). Behavior trees in robotics and ai: An introduction. *IEEE Robotics and Automation Magazine*, 28(2):74–85.

Dragan, A. D. and Srinivasa, S. S. (2013a). A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805.

Dragan, A. D. and Srinivasa, S. S. (2013b). A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805.

Engeberg, E. D. and Meek, S. (2008). Improved grasp force sensitivity for prosthetic hands through force-derivative feedback. *IEEE Transactions on Biomedical Engineering*, 55(2):817–821.

Falco, J., Hemphill, D., Kimble, K., Messina, E., Norton, A., Ropelato, R., and Yanco, H. (2020). Benchmarking protocols for evaluating grasp strength, grasp cycle time, finger strength, and finger repeatability of robot end-effectors. *IEEE Robotics and Automation Letters*, 5(2):644–651.

Fang, B., Wang, C., Sun, F., Chen, Z., Shan, J., Liu, H., Ding, W., and Liang, W. (2022). Simultaneous sEMG recognition of gestures and force levels for interaction with prosthetic hand. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:2426–2436.

Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., and Aszmann, O. C. (2014). The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):797–809.

Farina, D., Vujaklija, I., Sartori, M., Kapelner, T., Negro, F., Jiang, N., Bergmeister, K., Andalib, A., Principe, J., and Aszmann, O. C. (2017). Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation. *Nature biomedical engineering*, 1(2):0025.

Ficuciello, F. (2019). Synergy-based control of underactuated anthropomorphic hands. *TII*, 15(2):1144–1152.

Ficuciello, F., Palli, G., Melchiorri, C., and Siciliano, B. (2012). Planning and control during reach to grasp using the three predominant ub hand iv postural synergies. In *ICRA*, pages 2255–2260.

Ficuciello, F., Palli, G., Melchiorri, C., and Siciliano, B. (2014). Postural synergies of the UB Hand IV for human-like grasping. *RAS*, 62(4):515–527.

Ficuciello, F., Villani, A., Lisini Baldi, T., and Prattichizzo, D. (2021). A human gesture mapping method to control a multi-functional hand for robot-assisted laparoscopic surgery: The musha case. *Frontiers in Robotics and AI*, 8:741807.

Fleischer, C. and Hommel, G. (2008). A human–exoskeleton interface utilizing electromyography. *IEEE Transactions on Robotics*, 24(4):872–882.

Fougner, A., Stavdahl, Ø., Kyberd, P. J., Losier, Y. G., and Parker, P. A. (2012). Control of upper limb prostheses: Terminology and proportional myoelectric control—a review. *IEEE Transactions on neural systems and rehabilitation engineering*, 20(5):663–677.

Galassi, K., Caporali, A., and Palli, G. (2022). Cable detection and manipulation for DLO-in-hole assembly tasks. In *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 01–06. IEEE.

Galassi, K. and Palli, G. (2021). Robotic wires manipulation for switchgear cabling and wiring harness manufacturing. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 531–536. IEEE.

Gallagher, K. and Sambridge, M. (1994). Genetic algorithms: a powerful tool for large-scale nonlinear optimization problems. *Computers & Geosciences*, 20(7-8):1229–1236.

Gijsberts, A., Bohra, R., Sierra González, D., Werner, A., Nowak, M., Caputo, B., Roa, M. A., and Castellini, C. (2014). Stable myoelectric control of a hand prosthesis using non-linear incremental learning. *Frontiers in neurorobotics*, 8:8.

González Huarte, J., Ortiz de Zarate, M., and Ibarguren, A. (2023). Cad-based robot programming solution for wire harness manufacturing in aeronautic sector. *Robotics*, 12(5).

Haddadin, S. (2015). Safety evaluation of physical human–robot interaction via crash-testing. In *Proceedings of Robotics: Science and Systems (RSS)*.

Haddadin, S. (2016). Physical safety in robotics. In *Springer Handbook of Robotics*, pages 1285–1305. Springer.

Haddadin, S., Albu-Schaffer, A., and Hirzinger, G. (2008). Collision detection and reaction: A contribution to safe physical human–robot interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3356–3363.

Haggard, P. (2017). Sense of agency in the human brain. *Nature Reviews Neuroscience*, 18(4):196–207.

Hassan, M., Kadone, H., Ueno, T., Hada, Y., Sankai, Y., and Suzuki, K. (2018). Feasibility of synergy-based exoskeleton robot control in hemiplegia. *TNSRE*, 26(6):1233–1242.

Heisler, P., Utsch, D., Kuhn, M., and Franke, J. (2020). Optimization of wire harness assembly using human-robot-collaboration. *Procedia CIRP*, 97:260–265.

Howe, R. D. (1993). Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3):245–261.

Hu, J., Lu, Q., Fan, H., Xiao, Y., Zhou, Y., and Zhang, S. (2024). Unity 3d-based six-degree-of-freedom robotic arm virtual simulation teaching platform. In *2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)*, pages 265–270. IEEE.

Hu, S., Zhu, X., Wang, H., and Koren, Y. (2008). Product variety and manufacturing complexity in assembly systems and supply chains. *CIRP Annals*, 57(1):45–48.

Hu, Z., Yang, C., He, W., Li, Z., and He, S. (2017). Modeling and simulation of hand based on opensim and leap motion. In *2017 Chinese Automation Congress (CAC)*, pages 4844–4849. IEEE.

Huang, J., Li, G., Su, H., and Li, Z. (2021). Development and continuous control of an intelligent upper-limb neuroprosthesis for reach and grasp motions using biological signals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(6):3431–3441.

Huang, Z., Mun, Y.-J., Li, X., Xie, Y., Zhong, N., Liang, W., Geng, J., Chen, T., and Driggs-Campbell, K. (2023). Hierarchical intention tracking for robust human-robot collaboration in industrial assembly tasks. pages 9821–9828.

Huo, S., Duan, A., Li, C., Zhou, P., Ma, W., Wang, H., and Navarro-Alarcon, D. (2022). Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework. *IEEE Robotics and Automation Letters*, 7(2):5222–5229.

Hwang, J. Y., Kim, J. S., Lim, S. S., and Park, K. H. (2003). A fast path planning by path graph optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 33(1):121–129.

Iberall, T., Bingham, G. P., and Arbib, M. A. (1986). Opposition space as a structuring concept for the analysis of skilled hand movements.

International Organization for Standardization (2016). Iso/ts 15066: Robots and robotic devices — collaborative robots. Technical Specification.

Jain, A., Long, P., Villani, V., Kelleher, J. D., and Leva, M. C. (2024). Cobt: Collaborative programming of behaviour trees from one demonstration for robot manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12993–12999. IEEE.

Jiang, N., Englehart, K. B., and Parker, P. A. (2008). Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal. *IEEE transactions on Biomedical Engineering*, 56(4):1070–1080.

Jiang, N., Rehbaum, H., Vujaklija, I., Graimann, B., and Farina, D. (2013). Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):501–510.

Johannsmeier, L. and Haddadin, S. (2019). A proactive human–robot collaboration framework for industrial assembly. *Robotics and Computer-Integrated Manufacturing*, 57:1–12.

Johannsmeier, L. and Haddadin, S. (2020). Leveraging multimodal data for intuitive robot control in human-centered collaborative assembly. *Robotics and Computer-Integrated Manufacturing*, 61:101842.

Johannsmeier, L. and Haddadin, S. (2021). Human–robot collaboration and machine learning: A systematic review. *Robotics and Computer-Integrated Manufacturing*, 70:102041.

Kaboli, M., Yao, K., and Cheng, G. (2016). Tactile-based manipulation of deformable objects with dynamic center of mass. *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 752–757.

Katyara, S., Ficuciello, F., Caldwell, D. G., Siciliano, B., and Chen, F. (2023). Leveraging kernelized synergies on shared subspace for precision grasping and dexterous manipulation. *TCDS*, 15(4):2064–2076.

Khatib, O., Yeh, X., Brantner, G., Soe, B., Kim, B., Ganguly, S., Stuart, H., Wang, S., Cutkosky, M., Edsinger, A., et al. (2016). Ocean one: A robotic avatar for oceanic discovery. *IEEE Robotics & Automation Magazine*, 23(4):20–29.

Konak, A., Coit, D. W., and Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability engineering & system safety*, 91(9):992–1007.

Kumar, V. and Todorov, E. (2015). Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 657–663. IEEE.

Kyberd, P. J., Murgia, A., Gasson, M., Tjerks, T., Metcalf, C., Chappell, P. H., Warwick, K., Lawson, S. E., and Barnhill, T. (2009). Case studies to demonstrate the range of applications of the southampton hand assessment procedure. *British Journal of Occupational Therapy*, 72(5):212–218.

Laghi, M., Ajoudani, A., Catalano, M. G., and Bicchi, A. (2020). Unifying bilateral teleoperation and tele-impedance for enhanced user experience. *The International Journal of Robotics Research*, 39(4):514–539.

Lambora, A., Gupta, K., and Chopra, K. (2019). Genetic algorithm-a literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 380–384. IEEE.

Laudante, G., Mirto, M., Pennacchio, O., Galassi, K., Govoni, A., Pasquali, A., Pirozzi, S., Gosar, Ž., and Palli, G. (2025). Mechatronic integration of a dual-arm robotic system for wiring harness manufacturing. *IEEE/ASME Transactions on Mechatronics*.

Leardi, R. (2007). Genetic algorithms in chemistry. *Journal of Chromatography A*, 1158(1):226–233. Data Analysis in Chromatography.

Li, Z., Liu, J., Li, Z., Dong, Z., Teng, T., Ou, Y., Caldwell, D., and Chen, F. (2025). Language-guided dexterous functional grasping by llm generated grasp functionality and synergy for humanoid manipulation. *TASE*, 22:10506–10519.

Lv, N., Liu, J., and Jia, Y. (2022). Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations. *IEEE Transactions on Robotics*, 38(4):2341–2353.

Mac, T. T., Copot, C., Tran, D. T., and De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28.

Makris, S., Michalos, G., and Chryssolouris, G. (2017). Rampa: Robotic augmented reality for machine programming by demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3666–3671.

Mariano, M., Kuster, N., Tartufoli, M., and Zapparoli, L. (2024). How aging shapes our sense of agency. *Psychonomic Bulletin & Review*, 31(4):1714–1722.

Martinez, I. J. R., Mannini, A., Clemente, F., and Cipriani, C. (2020a). Online grasp force estimation from the transient EMG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(10):2333–2341.

Martinez, I. J. R., Mannini, A., Clemente, F., Sabatini, A. M., and Cipriani, C. (2020b). Grasp force estimation from the transient EMG using high-density surface recordings. *Journal of Neural Engineering*, 17(1):016052.

Mason, C. R., Gomez, J. E., and Ebner, T. J. (2001). Hand synergies during reach-to-grasp. *JN*, 86(6):2896–2910.

Mason, M. T. (2018). Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:1–28.

Matheson, E., Minto, R., Zampieri, E. G. G., Faccio, M., and Rosati, G. (2019). Human–robot collaboration in manufacturing applications: A review. *Robotics*, 8(4):100.

Meattini, R., Benatti, S., Scarcia, U., De Gregorio, D., Benini, L., and Melchiorri, C. (2018). An sEMG-based human–robot interface for robotic hands using machine learning and synergies. *IEEE Transactions on Components, Packaging and Manufacturing Technology*.

Meattini, R., Bernardini, A., Palli, G., and Melchiorri, C. (2022). sEMG-based minimally supervised regression using soft-dtw neural networks for robot hand grasping control. *IEEE Robotics and Automation Letters*, 7(4):10144–10151.

Meattini, R., Caporali, A., Bernardini, A., Palli, G., and Melchiorri, C. (2023a). Self-supervised regression of sEMG signals combining non-negative matrix factorization with deep neural networks for robot hand multiple grasping motion control. *IEEE Robotics and Automation Letters*.

Meattini, R., Suárez, R., Palli, G., and Melchiorri, C. (2023b). Human to robot hand motion mapping methods: Review and classification. *TRO*, 39(2):842–861.

Milstein, A., Ganel, T., Berman, S., and Nisky, I. (2018). Human-centered transparency of grasping via a robot-assisted minimally invasive surgery system. *IEEE Transactions on Human-Machine Systems*, 48(4):349–358.

Monforte, M. and Ficuciello, F. (2021). A reinforcement learning method using multifunctional principal component analysis for human-like grasping. *TCDS*, 13(1):132–140.

Monguzzi, A., Pelosi, M., Zanchettin, A. M., and Rocco, P. (2023). Tactile based robotic skills for cable routing operations. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3793–3799.

Moon, S., Sung, E., and Park, J. (2024). Kernel PCA-based hand synergy for efficient robot hand teleoperation using glove interface. In *ConfHumanoid23*, pages 17–23.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60.

Motamedi, M. R., Chossat, J.-B., Roberge, J.-P., and Duchaine, V. (2016). Haptic feedback for improved robotic arm control during simple grasp, slippage, and contact detection tasks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4894–4900. IEEE.

Murray, R. M. (2017). *A mathematical introduction to robotic manipulation*. CRC press.

Nguyen, H. G., Kuhn, M., and Franke, J. (2020). Manufacturing automation for automotive wiring harnesses. *Procedia CIRP*, 97:379–384.

Nguyen-Duc-Thanh, N., Lee, S., and Kim, D. (2012). Two-stage hidden markov model in gesture recognition for human robot interaction. *International Journal of Advanced Robotic Systems*.

Niemeyer, G., Preusche, C., Stramigioli, S., and Lee, D. (2016). Telerobotics. *Springer handbook of robotics*, pages 1085–1108.

Nowak, M., Vujaklija, I., Sturma, A., Castellini, C., and Farina, D. (2022). Simultaneous and proportional real-time myocontrol of up to three degrees of freedom of the wrist and hand. *IEEE Transactions on Biomedical Engineering*, 70(2):459–469.

Olbrich, S. and Lackinger, J. (2022). Manufacturing processes of automotive high-voltage wire harnesses: State of the art, current challenges and fields of action to reach a higher level of automation. *Procedia CIRP*, 107:653–660.

Osborn, L., Kaliki, R. R., Soares, A. B., and Thakor, N. V. (2016). Neuromimetic event-based detection for closed-loop tactile feedback control of upper limb prostheses. *IEEE Transactions on Haptics*, 9(2):196–206.

Palli, G., Melchiorri, C., Vassura, G., Scarcia, U., Moriello, L., Berselli, G., Cavallo, A., Maria, G. D., Natale, C., Pirozzi, S., May, C., Ficuciello, F., and Siciliano, B. (2014). The DEXMART hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping. *IJRR*, 33(5):799–824.

Palomba, I., Gualtieri, L., Rojas, R., Rauch, E., Vidoni, R., and Ghedin, A. (2021). Mechatronic re-design of a manual assembly workstation into a collaborative one for wire harness assemblies. *Robotics*, 10(1).

Pasquali, A., Alunni, M. M., Chiaravalli, D., and Palli, G. (2024). Stiffness-aware genetic algorithm for robotic path finding optimization. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 942–947. IEEE.

Pasquali, A., Bargellini, D., Meattini, R., Crevillén, D., Egle, F., Gentile, C., Saetta, G., Gruppioni, E., Castellini, C., and Palli, G. (2026a). iHandVR: An immersive virtual reality platform for functional testing with prosthetic hand control. Submitted to IEEE Transactions on Human-Machine Systems.

Pasquali, A., Galassi, K., and Palli, G. (2023). A fast score-based method for robotic task-free point-to-point path learning. In *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1159–1164. IEEE.

Pasquali, A., Meattini, R., Bernardini, A., Govoni, A., Laudante, G., Melchiorri, C., and Palli, G. (2026b). Grip-force-aware human-in-the-loop control and programming of robotic grasping via probabilistic shared autonomy. Submitted to IEEE Transactions on Mechatronics.

Pecora, F. et al. (2018). Skiros2: A skill-based robot control platform for industrial applications. *Autonomous Robots*, 42(6):1175–1190.

Pfeiffer, S. and Bdiwi, M. (2020). M2o2p: A multimodal programming framework for collaborative robots. *Sensors*, 20(16):4521.

Piggott, L., Wagner, S., and Ziat, M. (2016). Haptic neurorehabilitation and virtual reality for upper limb paralysis: a review. *Critical Reviews™ in Biomedical Engineering*, 44(1-2).

Porges, O., Connan, M., Henze, B., Gigli, A., Castellini, C., and Roa Garzon, M. A. (2019). A wearable, ultralight interface for bimanual teleoperation of a compliant, whole-body-controlled humanoid robot. In *2019 International Conference on Robotics and Automation, ICRA 2019*. IEEE.

Prattichizzo, D., Malvezzi, M., Gabiccini, M., and Bicchi, A. (2013). On motion and force controllability of precision grasps with hands actuated by soft synergies. *IEEE transactions on robotics*, 29(6):1440–1456.

Prattichizzo, D. and Trinkle, J. C. (2016). Grasping. *Springer handbook of robotics*, pages 955–988.

Pylatiuk, C., Kargov, A., and Schulz, S. (2006). Design and evaluation of a low-cost force feedback system for myoelectric prosthetic hands. *JPO: Journal of Prosthetics and Orthotics*, 18(2):57–61.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Raspopovic, S., Capogrosso, M., Petrini, F. M., Bonizzato, M., Rigosa, J., Di Pino, G., Carpaneto, J., Controzzi, M., Boretius, T., Fernandez, E., et al. (2014). Restoring natural sensory feedback in real-time bidirectional hand prostheses. *Science translational medicine*, 6(222):222ra19–222ra19.

Raveh, E., Friedman, J., and Portnoy, S. (2018). Evaluation of the effects of adding vibrotactile feedback to myoelectric prosthesis users on performance and visual attention in a dual-task paradigm. *Clinical Rehabilitation*, 32(10):1308–1316.

Reddy, S., Dragan, A. D., Levine, S., and Abbeel, P. (2018). Shared autonomy via deep reinforcement learning. In *Proceedings of Robotics: Science and Systems (RSS)*.

Regoli, M., Pattacini, U., Metta, G., and Natale, L. (2016). Hierarchical grasp controller using tactile feedback. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 387–394. IEEE.

Roberge, V., Tarbouchi, M., and Labonte, G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141.

Rodríguez-Fernández, A., den Berg, A. v., Cucinella, S. L., Lobo-Prat, J., Font-Llagunes, J. M., and Marchal-Crespo, L. (2024). Immersive virtual reality for learning exoskeleton-like virtual walking: a feasibility study. *Journal of neuroengineering and rehabilitation*, 21(1):195.

Romeo, R. A., Lauretti, C., Gentile, C., Guglielmelli, E., and Zollo, L. (2021). Method for automatic slippage detection with tactile sensors embedded in prosthetic hands. *IEEE Transactions on Medical Robotics and Bionics*, 3(2):485–497.

Romero, J., Feix, T., Ek, C. H., Kjellström, H., and Kragic, D. (2013). Extracting postural synergies for robotic grasping. *TRO*, 29(6):1342–1352.

Romero, J., Feix, T., Kjellström, H., and Kragic, D. (2010). Spatio-temporal modeling of grasping actions. In *IROS*, pages 2103–2108.

Román Ibáñez, V., Pujol, F., García Ortega, S., and Sanz Perpiñán, J. (2021). Collaborative robotics in wire harnesses spot taping process. *Computers in Industry*, 125:103370.

ROS-Industrial Consortium (2020). Ros-industrial. https://rosindustrial.org.

Rozo, L., Silverio, J., Calinon, S., and Caldwell, D. G. (2016). Learning controllers for reactive and proactive behaviors in human–robot collaboration. *IEEE Transactions on Robotics*, 32(1):1–14.

Santello, M., Bianchi, M., Gabiccini, M., Ricciardi, E., Salvietti, G., Prattichizzo, D., Ernst, M., Moscatelli, A., Jörntell, H., Kappers, A. M., et al. (2016). Hand synergies: integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Physics of life reviews*, 17:1–23.

Santello, M., Flanders, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. *The Journal of Neuroscience*, 18(23):10105–10115.

Santello, M., Flanders, M., and Soechting, J. F. (2002). Patterns of hand motion during grasping and the influence of sensory guidance. *JN*, 22(4):1426–1435.

Santina, C. D., Piazza, C., Grioli, G., Catalano, M. G., and Bicchi, A. (2018). Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. *TRO*, 34(5):1141–1156.

Sartori, M., Llyod, D. G., and Farina, D. (2016). Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies. *IEEE transactions on biomedical engineering*, 63(5):879–893.

Scheme, E. and Englehart, K. (2011). Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6).

Selvaggio, M., Moccia, R., Ficuciello, F., Siciliano, B., et al. (2019). Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3617–3623. IEEE.

She, Y., Wang, S., Dong, S., Sunil, N., Rodriguez, A., and Adelson, E. (2021). Cable manipulation with a tactile-reactive gripper. *International Journal of Robotics Research*, 40(12-14):1385 – 1401.

Shukla, A. and Billard, A. (2012). Coupled dynamical system based arm–hand grasping model for learning fast adaptation strategies. *RAS*, 60(3):424–440.

Si, Z., Zhang, K., Kroemer, O., and Temel, F. Z. (2024). Deltahands: A synergistic dexterous hand framework based on delta robots. *RAL*, 9(2):1795–1802.

Silver-Thorn, B., Steege, J. W., and Childress, D. S. (1996). A review of prosthetic interface stress investigations. *Journal of rehabilitation research and development*.

Son, D. (2022). Grasping as inference: Reactive grasping in heavily cluttered environment. *IEEE Robotics and Automation Letters*, 7(3):7193–7200.

Srinivas, M. and Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667.

Starke, J., Eichmann, C., Ottenhaus, S., and Asfour, T. (2018). Synergy-based, data-driven generation of object-specific grasps for anthropomorphic hands. In *ConfHumanoid18*, pages 327–333.

Starke, J., Eichmann, C., Ottenhaus, S., and Asfour, T. (2020). Human-inspired representation of object-specific grasps for anthropomorphic hands. *IJHR*, 17(02):2050008.

Starke, J., Keller, M., and Asfour, A. (2021). Temporal force synergies in human grasping. In *IROS*, pages 3963–3970.

Stella, F., Achkar, M. M., Della Santina, C., and Hughes, J. (2025). Synergy-based robotic quadruped leveraging passivity for natural intelligence and behavioural diversity. *NMI*, pages 1–14.

Stiefelhagen, R. et al. (2019). Gesture- and speech-guided robot teleoperation in industrial environments. *IEEE Transactions on Cognitive and Developmental Systems*, 11(2):204–216.

Tam, S., Boukadoum, M., Campeau-Lecours, A., and Gosselin, B. (2021). Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning. *Scientific Reports*, 11(1):11275.

Thomas, N., Miller, A. J., Ayaz, H., and Brown, J. D. (2023). Haptic shared control improves neural efficiency during myoelectric prosthesis use. *Scientific Reports*, 13(1):484.

Thomas, N., Ung, G., McGarvey, C., and Brown, J. D. (2019). Comparison of vibrotactile and joint-torque feedback in a myoelectric upper-limb prosthesis. *Journal of NeuroEngineering and Rehabilitation*, 16(1):70.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.

Trommnau, J., Kühnle, J., Siegert, J., Inderka, R., and Bauernhansl, T. (2019). Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends. *Procedia CIRP*, 81:387–392.

Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266.

Wang, F., Meattini, R., Pasquali, A., Lan, H., Song, G., Song, A., and Melchiorri, C. (2026). Improving synergy subspace reach-to-grasp trajectories via local probabilistic nonlinear latent representations. Submitted to IEEE Robotics and Automation Letters.

Wen, R., Yuan, K., Wang, Q., Heng, S., and Li, Z. (2020). Force-guided high-precision grasping control of fragile and deformable objects using semg-based force prediction. *IEEE Robotics and Automation Letters*, 5(2):2762–2769.

Witteveen, H. J., Luft, F., Rietman, J. S., and Veltink, P. H. (2013). Stiffness feedback for myoelectric forearm prostheses using vibrotactile stimulation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1):53–61.

Xu, K., Liu, H., Du, Y., and Zhu, X. (2016). A comparative study for postural synergy synthesis using linear and nonlinear methods. *IJHR*, 13(03):1650009.

Yu, M., Lv, K., Zhong, H., Song, S., and Li, X. (2023). Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach. *IEEE Transactions on Robotics*, 39(1):417 – 436.

Zaidi, L., Corrales, J. A., Bouzgarrou, B. C., Mezouar, Y., and Sabourin, L. (2017). Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems*, 95:196–206.

Zhu, J., Cherubini, A., Dune, C., Navarro-Alarcon, D., Alambeigi, F., Berenson, D., Ficuciello, F., Harada, K., Kober, J., Li, X., Pan, J., Yuan, W., and Gienger, M. (2022). Challenges and outlook in robotic manipulation of deformable objects. *IEEE Robotics & Automation Magazine*, 29(3):67–77.

Zhuang, K. Z., Sommer, N., Mendez, V., Aryan, S., Formento, E., D'Anna, E., Artoni, F., Petrini, F., Granata, G., Cannaviello, G., et al. (2019). Shared human–robot proportional control of a dexterous myoelectric prosthesis. *Nature Machine Intelligence*, 1(9):400–411.