



Politecnico di Torino

DRIM

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

Administrative Headquarters: Università di Genova

Design and Control of Compliant Robotic Systems for Hand Rehabilitation and Tendon-Driven Continuum Robots

by

Mohammad Jabari

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

Prof. Giuseppe Quaglia
Prof. Carmen Visconte
Prof. Med Amine Laribi
Antonio Sgorbissa

Supervisor
Supervisor
Supporting Supervisor
Head of the PhD program

Thesis Jury:

Prof. Victor Petuya
Prof. Habil. Erwin-Christian Lovasz
Prof. Walter Franco

External examiner
External examiner
Internal examiner

Politecnico di Torino

September 2026

I dedicate this thesis to my wife, Somayeh, and to my family,
for their constant love, patience, and support throughout this journey.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis 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 thesis 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.

Mohammad Jabari

July 2026

Acknowledgements

This thesis is part of the PNRR-NGEU project, funded by the Italian Ministry of University and Research (MUR) under DM 351/2022.

I would like to sincerely thank Prof. Giuseppe Quaglia, Prof. Carmen Visconte, and Prof. Med Amine Laribi for their guidance, support, and encouragement throughout my doctoral studies. Their insight and experience have been invaluable to the development of this research and to my growth as a researcher.

I am also grateful to the research teams at Politecnico di Torino and the CoBRA Team at University of Poitiers for their feedback, collaboration, and support during this multidisciplinary work. This manuscript and the resulting works would not have been possible without their contribution and support.

Finally, I would like to express my deepest gratitude to my wife, Somayeh, for her patience, love, and constant encouragement throughout this journey. I am also thankful to my family and friends for their continuous support and encouragement during my doctoral studies.

Mohammad Jabari

Turin

July 2026

Abstract

This thesis investigates the design and control of compliant robotic systems for human-centred healthcare, with a dual focus on wearable hand rehabilitation devices and tendon-driven continuum robots (TDCRs) for minimally invasive interventions. The aim is to develop robotic systems that remain mechanically feasible under realistic practical constraints while maintaining compatibility with human anatomy.

The first part focuses on hand rehabilitation. A finger exoskeleton architecture is proposed for the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joints based on Remote Centre of Motion (RCM) mechanisms. The linkage generates virtual centres aligned with anatomical joint axes. Most of the mechanism remains within the hand profile, helping keep the device compact and wearable. To provide adjustable resistance, a pneumatic variable-stiffness layer is developed using 3D-printed inflatable chambers integrated into a glove. The chamber design was gradually improved through prototyping, from early Flexojoint concepts to a compact pillow-shaped geometry. Experiments were conducted to quantify the influence of bending angle on internal pressure, reaction force, and resistive torque across a 0° – 45° range under various initial inflation pressures in a sealed pneumatic circuit. These experiments show pressure-tunable torque–angle profiles that can support different rehabilitation stages. The second part focuses on the design, planning, and control of tendon-driven continuum robots under practical constraints. First, a model-based framework is developed to analyse static feasibility using a discretized PCC representation and a kineto-static formulation that accounts for tendon forces, gravity, friction, backbone elasticity, and external loads. This framework is subsequently used to analyse the feasible static workspace and to optimise tendon-routing parameters so as to maximise workspace feasibility while minimising mechanical effort. Based on these results, two planning strategies are introduced: a hybrid inverse-kinematics scheme combining supervised learning with constrained refinement, and an offline capability-map framework that stores feasible voxel-orientation targets together with dexterity information for runtime selection. Finally, prescribed-time safe and fault-tolerant control strategies are developed, including observer-based estimation and learning-assisted compensation under uncertainties, delays, actuator faults, and input saturation.

Table of contents

List of figures	ix
List of tables	xiv
1 Introduction and Thesis Outline	1
1.1 Contributions of the Thesis	2
1.2 Research Methods	2
1.3 Thesis Outline and Chapter Synthesis	3
2 Biomechanical Foundations and State of the Art in Medical Robotics	8
2.1 Hand Rehabilitation: Anatomy, Impairment, and Technology	8
2.1.1 Anatomy and Biomechanics of the Finger	9
2.1.2 Causes of Hand Impairment and Clinical Pathways	9
2.1.3 Robot-Assisted Rehabilitation: Representative Systems and Design Trade-offs	9
2.2 Fundamentals of Tendon-Driven Continuum Robots	10
2.2.1 Kinematic and Static Modelling	10
2.2.2 Workspace Analysis: Feasible Static Workspace and Capability Maps	12
2.2.3 Advanced Control and Learning-Based Approaches	12
2.3 Chapter Summary	13
I Section Two	14
3 Hand Rehabilitation Systems: RCM Exoskeleton and Inflatable Chambers	15
3.1 Anatomical Basis and Geometric Parameters of the Finger	15
3.2 Problem Statement and Design Objectives	17
3.3 Design Concept and Kinematic Principle of the RCM Exoskeleton	18

3.4	Development of Compliant Variable-Stiffness Modules	19
3.4.1	From Flexojoint to Pillow-Shaped Chambers	19
3.4.2	Glove Integration and Coupling to the RCM Layer	20
3.5	Experimental Characterization	21
3.5.1	Experimental Setup and Procedure	21
3.5.2	Pressure Evolution	22
3.5.3	Force Evolution	23
3.5.4	Torque Response	23
3.6	Rehabilitation Implications and Limitations	25
3.7	Summary	25
4	Model-Based Foundations for Static Feasibility and Design Optimization of Tendon-Driven Continuum Robots	26
4.1	Motivation and Scope of the Chapter	26
4.2	Robot Architecture and Modelling Assumptions	28
4.2.1	Mechanical Structure and Tendon Routing	28
4.2.2	Modelling Assumptions	30
4.3	Kinematic Modelling of TDCR	31
4.3.1	Discretized PCC Representation	31
4.3.2	Subsegment Transformation and Forward Kinematics	32
4.4	Kineto-Static Modelling Framework	33
4.4.1	Loads and Contributions at Vertebra Level	33
4.4.2	Friction Modelling and Tension Propagation	34
4.4.3	Backbone Elasticity	35
4.4.4	Recursive Force and Moment Equilibrium	35
4.5	Study A: Feasible Static Workspace and Tendon-Force Optimization	36
4.5.1	Definition	36
4.5.2	Scalar Metric and Optimization Problem	37
4.5.3	Numerical Results	37
4.6	Study B: Multi-Objective Dimensional Synthesis of Tendon Routing	39
4.6.1	Design Variables	40
4.6.2	Objective Functions and Performance Indicators	40
4.6.3	Geometric Bounds and Non-Interference Constraint	41
4.6.4	Optimization Procedure	42
4.6.5	Results and Design Guidelines	42

4.7	Chapter Summary	45
5	From Hybrid Task-Space Planning to Offline Capability-Map for Tendon-Driven Continuum Robots	47
5.1	Motivation and Scope of the Chapter	47
5.2	Task-Space Trajectory and Discretisation	48
5.3	Tendon Length Variation Profiles	50
5.4	Offline IK Data Generation for Supervised Learning	51
5.5	Supervised Learning Model and Training Setup	52
5.6	Hybrid Online Inverse-Kinematics Pipeline	53
5.7	Simulation Results and Performance Evaluation	54
5.8	Motivation for Offline Capability Mapping	57
5.9	Capability-Map Database Representation of the Studied TDCR	58
5.10	Offline Construction of the Capability Database	60
5.10.1	PSO-Based Inverse Kinematics and Acceptance Filtering	61
5.10.2	Actuation Mapping	63
5.11	Offline Differential Kinematics and Dexterity Evaluation	64
5.12	Database-Driven Bi-Objective Configuration Selection	66
5.13	Offline CAP Analysis and Configuration Selection Results	68
5.13.1	IK Solver Quality	68
5.13.2	Distribution of the Adopted Dexterity Metric	70
5.13.3	Bi-Objective Selection Analysis	74
5.14	Experimental Validation	77
5.15	Chapter Summary	81
6	Prescribed-Time Fault-Tolerant Control of Tendon-Driven Continuum Robots	83
6.1	Motivation and Scope of the Chapter	83
6.2	Mathematical Preliminaries and Dynamic Representation	86
6.2.1	Bending Coordinates and Problem Setting	86
6.2.2	Dynamic Model Used for Control Design	87
6.2.3	Control Objective	88
6.3	Prescribed-Time Safe Tracking via Monotone Tube Boundaries	88
6.3.1	Monotone Tube Boundary Formulation	88
6.3.2	Prescribed-Time Sliding Structure	90
6.3.3	Learning-Enhanced Optimal Compensation	91
6.3.4	Closed-Loop Stability Result	93

6.4	Numerical Evaluation of the Nominal Controller	94
6.4.1	Simulation Setting	94
6.5	Observer-Based Prescribed-Time Fault-Tolerant Formulation	100
6.5.1	Fault-Affected Model and Augmented Representation	101
6.5.2	Prescribed-Time Descriptor Observer	102
6.5.3	Composite Deep-Learning-Based Fault-Tolerant Control Law	106
6.6	Numerical Evaluation of the Fault-Tolerant Controller	111
6.6.1	Simulation Setting	112
6.6.2	Step-Reference Tracking	113
6.6.3	Sinusoidal Tracking under Delays and Faults	116
6.6.4	Multiple-Fault Resilience	117
6.7	Chapter Summary	120
7	Conclusions and Further Research	122
7.1	Conclusions	122
7.2	Further Research	124
	References	126

List of figures

3.1	Anatomical details of the hand skeleton.	16
3.2	Concept overview of the hybrid rehabilitation system. (a) Index finger anatomy and joint locations (DIP, PIP, MCP). (b) CAD model illustrating the RCM-based linkage and the placement of inflatable chambers.	17
3.3	RCM-based finger mechanism and its coupling with the sealed pneumatic chamber system used to generate pressure-dependent resistance during flexion. (a) Side view with dimensions in millimetres. (b) Rotation mechanism. Numbers correspond to: 1 = forward chamber, 2 = backward chamber, 3 = thimble, 4 = air tube, 5 = index finger, 6 = pneumatic system, 7 = pressure sensor, 8 = hand lever valve, 9 = air supply, 10 = air regulator.	19
3.4	Chamber design evolution and final geometry. (a) Early Flexojoint prototype. (b) Intermediate concept. (c) Final pillow-shaped chamber with rigid plates and TPU inflatable cavity. (d) Distal thimble integration.	20
3.5	System integration. (a) CAD layout showing chamber placement near MCP and PIP. (b) Assembly on the hand with the exoskeleton mounted over the glove. (c) Prototype with pneumatic lines connected.	21
3.6	Experimental setup used to characterise the pillow-shaped chambers, including the bending fixture, sealed pneumatic circuit, pressure sensor, manual valve, and dynamometer.	22
3.7	Pressure evolution $p(\theta, p_{\text{init}})$ for forward and backward chambers for different initial pressures.	23
3.8	Measured bending force in the forward and backward chambers across bending angles for different initial pressures.	24
3.9	Resistive torque $\tau(\theta, p_{\text{init}})$ at MCP and PIP joints for different initial pressures.	24

4.1	Diagram of the actuator unit and drive mechanism. (a) Driving cable arrangement, with tendons 1–4 for Segment 1 shown in red and tendons 1–4 for Segment 2 shown in blue. (b) Actuator unit with motors M1–M4 arranged with a 45° offset. (c) Control unit. (d) Motor with mounted pulley and protective guard. (e) Pulley mechanism detail.	29
4.2	Overview of the two-segment tendon-driven continuum robot prototype and its structural layout. (a) Experimental setup with actuation unit and control box. (b) Four-tendon arrangement on each segment and inter-segment angular offset α . (c) Tendon-routing geometry. (d) Vertebra and compliant joint. (e) CAD model of the proposed structure.	31
4.3	PCC parametrisation of a single backbone subsegment between two consecutive vertebra frames.	32
4.4	Free-body diagram of vertebra i and the adjacent backbone subsegment, showing tendon forces, gravity, and internal resultants.	34
4.5	Feasible static workspace obtained from 100 randomly sampled tendon-force vectors.	38
4.6	The trend of objective function changes during tendon-force optimization.	38
4.7	Optimal equilibrium configuration under vertical tip load.	39
4.8	(a) Four-tendon arrangement on the vertebra for both segments. (b) Geometric constraint ensuring minimum tendon spacing.	40
4.9	(a) Pareto front diagram, 14th iteration. (b) Pareto front diagram, 61st iteration.	43
4.10	Frequency histogram of the data as a function of α angle divisions.	44
4.11	Average α angle for the optimal samples across the angular divisions.	44
4.12	Variance in the α angle within each division.	44
4.13	Average radial distance of the tendons from the center of the vertebra for each segment.	44
4.14	Variance in radial tendon distances for each segment.	44
4.15	Inverse work across the range of α angle variations.	44
4.16	Variance of inverse work across the range of α angle variations.	45
4.17	Static feasible workspace across the range of α angle variations.	45
4.18	Variance of static feasible workspace across the range of α angle variations.	45
5.1	Geometric interpretation of tendon-length variation. (a) Deformation between two consecutive vertebrae. (b) Tendon placement in the cross-section. (c) Projection of the tendon position onto the bending direction.	50

5.2	Circular trajectory tracking using the hybrid inverse kinematics framework. (a) HybMLP versus target path. (b) HybDNN versus target path. (c) Three-dimensional TDCR configuration at sample 617 out of 1000 along the trajectory. The dashed black curve denotes the target path, while the color-coded curve represents the HybDNN trajectory, with colour encoding progression along the path.	55
5.3	Helical trajectory tracking using the hybrid inverse kinematics framework. (a) HybMLP versus target path. (b) HybDNN versus target path. (c) Three-dimensional TDCR configuration at sample 969 out of 1000 along the trajectory. The dashed black curve denotes the target path, while the color-coded curve represents the HybDNN trajectory.	55
5.4	Tendon length variations in Segment 1 during circular trajectory execution, four tendons plotted over waypoint index.	56
5.5	Tendon length variations in Segment 2 during circular trajectory execution, four tendons plotted over waypoint index.	56
5.6	Workspace and CAP sampling strategy: (a) sampled reachable workspace of the two-segment TDCR obtained through forward-kinematic sampling; (b) TDCR model with the selected distal region of interest; (c) isolated view of the selected ROI and its voxelised representation; (d) lower-hemisphere sampling of candidate tool-axis directions for one representative voxel; and (e) bottom-view projection of the 300 distributed candidate directions. . . .	62
5.7	Reachability classification of the selected ROI: 88 reachable voxels, green, 48.9%, and 92 unreachable voxels, red, 51.1%, under the adopted backbone-length and curvature constraints.	69
5.8	Representative voxel-level IK result showing the target frame and the solver-found pose inside the spherical neighbourhood associated with the selected voxel.	69
5.9	Representative solver-found TDCR configuration. (a) 3D reconstructed backbone with the selected voxel and local frame. (b) $x - z$ projection. (c) $y - z$ projection.	71
5.10	Global distributions of the 6D manipulability measure w_{6D} and the positional manipulability measure w_p over all feasible CAP entries.	72
5.11	Log-scale relationship between the 6D manipulability measure w_{6D} and the positional manipulability measure w_p over all feasible CAP entries.	72

5.12	Distribution of the 6D manipulability measure w_{6D} over feasible CAP entries within selected workspace voxels.	73
5.13	Ranking of workspace voxels based on the mean 6D manipulability measure w_{6D} . (a) Ten voxels with the highest mean w_{6D} . (b) Ten voxels with the lowest mean w_{6D}	73
5.14	Voxel-level residual-dexterity trade-off for Voxel 177. The axes show the combined pose residual f_1 and inverse 6D manipulability f_2 . Gray points denote feasible CAP entries, red markers indicate the Pareto set, and the blue star marks the representative entry selected by the normalized ideal-point criterion.	75
5.15	Voxel-level residual–dexterity trade-off for Voxel 164, showing a sparser feasible CAP distribution relative to Voxel 177. Axes, markers, and the normalized ideal-point selection criterion follow Figure 5.14.	75
5.16	PCC parameters of the representative CAP entry selected for Voxel 177. The disk-wise curvature, bending-plane angle, and twist profiles define the internal TDCR configuration identified by the residual-dexterity Pareto/ideal-point criterion.	76
5.17	PCC parameters of the representative CAP entry selected for Voxel 164. The resulting internal TDCR configuration is distinct from that selected for Voxel 177. Parameter definitions and selection criterion follow Figure 5.16.	76
5.18	Hardware validation of the selected CAP entry. (a) TDCR–OptiTrack experimental setup. (b) Executed configuration for Voxel ID 151 and Sample ID 45141. (c) Target, solver-found, and experimentally measured poses within the selected voxel.	78
5.19	Spatial comparison of the CAP target, solver-found CAP pose, and experimentally measured start/end poses in Voxel 151.	79
5.20	Position and orientation errors of the experimentally executed CAP samples associated with Voxel 151, reported with respect to both the nominal CAP target and the solver-found pose.	79
5.21	Experimental error distributions for Voxel 151. Position and orientation errors are summarized with respect to the nominal CAP target and the stored solver-found pose.	80
6.1	Two Segment Continuum Robot Structure.	87

6.2	Block diagram of the proposed prescribed-time monotone-tube tracking controller.	92
6.3	Comparison of continuum robot bending angles tracking under case 1. . . .	96
6.4	Comparison of continuum robot bending angles tracking error under case 1.	96
6.5	Comparison of control inputs under case 1.	97
6.6	Evolution of Actor and Critic NN weights in case 1.	97
6.7	Comparison of continuum robot bending angles tracking under case 2. . . .	98
6.8	Comparison of continuum robot bending angles tracking under case 2. . . .	98
6.9	Comparison of control inputs under case 2.	99
6.10	Comparison of continuum robot bending angles tracking error under case 3.	100
6.11	Comparison of continuum robot bending angles tracking under case 3. . . .	100
6.12	Block diagram of the proposed composite adaptive optimal control.	109
6.13	Comparison of continuum robot bending angles tracking under step signal. .	114
6.14	Comparison of continuum robot bending angles tracking error under step signal.	114
6.15	Comparison of control inputs under step signal.	115
6.16	Dynamic variables of the continuum robot system and their estimated values under step signal.	115
6.17	Comparison of continuum robot bending angles tracking under sinusoidal reference signal.	116
6.18	Comparison of continuum robot bending angles tracking error under sinusoidal signal.	116
6.19	Comparison of control inputs under sinusoidal signal.	117
6.20	System states and observer estimates under sinusoidal references.	118
6.21	Comparison of continuum robot bending angles tracking under multiple faults.	119
6.22	System states and observer estimates under multiple faults.	119
6.23	Sensor faults estimation.	120

List of tables

2.1	Representative wearable hand rehabilitation systems and the proposed hybrid concept.	11
3.1	Mean long-finger phalanx lengths in millimetres, averaged across sex and side.	16
3.2	Subject-specific phalanx lengths used for prototype parameterisation in millimetres.	16
4.1	Geometric and mechanical parameters of the two-segment TDCR.	29
4.2	Statistical summary of Pareto-optimal designs grouped by α -range across all optimisation runs.	43
5.1	Summary of supervised learning architectures and training setup.	53
5.2	Computational performance of neural and hybrid inverse-kinematics models under the Lock Y-axis constraint ($N = 1000$ samples).	56
5.3	Representative raw IK outcomes for five sampled voxel–orientation targets before the CAP acceptance filtering.	70
6.1	Main parameters used in the observer-based fault-tolerant simulations. . . .	113

Chapter 1

Introduction and Thesis Outline

In recent decades, robotics has expanded from structured industrial environments to human-centred applications in healthcare, rehabilitation, and minimally invasive medicine. In these contexts, robots must operate in close proximity to the human body while respecting anatomical constraints and ensuring safe, predictable interaction. This shift places greater emphasis on compact integration, kinematic compatibility, and safety, without compromising performance.

In hand rehabilitation, restoring functional finger motion requires devices that can guide joint movement while limiting unwanted interface loads. Rigid exoskeletons can provide accurate motion guidance and repeatability, yet their performance is often limited by joint misalignment and relative motion at the finger–device interface during repeated flexion–extension. Soft robotic gloves offer lightweight and compliant interaction, but they often provide less precise joint-level guidance and less repeatable control of the resistive torque needed for strengthening exercises. These trade-offs motivate hybrid designs that combine kinematic compatibility with adjustable resistance in a compact wearable form.

In minimally invasive interventions, tendon-driven continuum robots (TDCRs) offer high dexterity and compliance for navigating constrained anatomical pathways. However, clinically useful operation requires more than geometric reachability. Configurations must remain feasible under tendon-force limits and external loads, and high-dimensional inverse kinematics must be solved while enforcing actuation and safety constraints. These requirements motivate methods based on feasible static workspace analysis and capability mapping, together with planning and control strategies that combine physics-based constraints with data-driven elements.

This thesis addresses both domains through a common perspective on compliant robotic systems, combining mechanism design, experimental characterisation, modelling, learning-

assisted planning and optimisation, and robust control to improve anatomical compatibility, feasibility under constraints, and performance in safety-critical tasks.

1.1 Contributions of the Thesis

In rehabilitation robotics, this thesis proposes a compact RCM-based finger exoskeleton for the MCP and PIP joints, generating virtual centres aligned with the anatomical joint axes while keeping most components within the hand profile to preserve wearability. To complement kinematic guidance with tunable resistance, glove-integrated 3D-printed pneumatic chambers are also developed. The chamber design is progressively refined from early Flexojoint prototypes to a pillow-shaped geometry, and experiments characterise the relationships among bending angle, internal pressure, reaction force, and resistive torque in a sealed pneumatic circuit under different initial inflation pressures. The results demonstrate pressure-tunable torque–angle profiles suitable for different rehabilitation modalities.

In continuum robotics, the thesis develops modelling and optimisation tools for design and capability assessment under practical constraints. A kineto-static framework is used to compute the Feasible Static Workspace (FSW) under external forces and torques, and an orientation-aware capability mapping approach is introduced to characterise feasible reach and attainable poses. Building on these tools, a hybrid inverse-kinematics framework is proposed in which supervised learning provides fast initial estimates and constrained optimisation refines them while enforcing tendon limits and feasibility constraints, with an orientation-aware step improving orientation accuracy. Finally, prescribed-time fault-tolerant control strategies are developed and analysed, and their behaviour is assessed in simulation under uncertainties, actuator faults, delays, and input saturation.

1.2 Research Methods

The research follows a structured workflow combining literature-based requirements, mechanism design, experimental characterisation, modelling, and learning-based methods. For the rehabilitation part, anatomical and functional requirements are translated into a compact RCM linkage whose geometry is parameterised using finger measurements and then verified through CAD-based inspection of the assembled mechanism. Pneumatic modules are developed through iterative prototyping, and their behaviour is experimentally characterised by measuring internal pressure and reaction force during quasi-static bending at different initial

inflation pressures in a sealed circuit; resistive torque profiles are computed using known lever arms.

For the TDCR part, kinematic and kineto-static models are implemented to support optimisation-based analyses of workspace and capability under tendon-force and loading constraints. Large-scale datasets are generated through structured task-space sampling. These datasets are built using forward kinematics together with constrained inverse-kinematics procedures under feasibility and orientation constraints. Supervised neural networks are trained to approximate inverse pose-to-configuration mappings, and their outputs are refined through constrained optimisation to enforce physical limits and feasibility. Control strategies are developed and analysed using Lyapunov-based tools and validated through numerical simulations under representative uncertainty and fault scenarios.

1.3 Thesis Outline and Chapter Synthesis

The thesis is organised into two main parts, addressing wearable hand rehabilitation and tendon-driven continuum robots. Although these application domains differ in scale and clinical purpose, they are connected by a common methodological perspective: compliance is considered as a system-level requirement, rather than a simple mechanical property. In both domains, this requirement must be addressed together with the anatomical, mechanical, actuation, planning, and execution constraints that determine whether a robotic system is practically usable. In the rehabilitation part, this perspective is addressed through anatomical compatibility and pressure-dependent mechanical interaction. In the continuum-robot part, it is extended to static feasibility, constrained planning, capability-aware configuration selection, and safe fault-tolerant control. The relationship between the two parts is therefore conceptual and methodological rather than a direct transfer of models or experimental data. The chapters are summarised below in terms of their objectives, methodologies, validation procedures, and main outcomes.

Chapter 2 – Biomechanical Foundations and State of the Art in Medical Robotics

Objective: Establish the anatomical, clinical, and technological requirements that guide the developments of the thesis.

Methodology: Review finger biomechanics, rehabilitation pathways, and representative rigid and soft wearable systems, followed by a review of TDCR kinematics, kineto-static modelling, feasible static workspace, capability mapping, learning-assisted planning, and robust control.

Validation basis: The chapter does not introduce a separate experimental study; instead, the requirements are supported through the reviewed literature and the comparative analysis of representative rehabilitation systems.

Main outcomes: The chapter identifies the need to combine joint-level kinematic compatibility with adjustable resistance in hand rehabilitation and to distinguish geometric reachability from physically feasible and functionally useful configurations in continuum robotics. These requirements define the design objective of Chapter 3 and the modelling, planning, and control developments of Chapters 4–6.

Chapter 3 – Hand Rehabilitation Systems: RCM Exoskeleton and Inflatable Chambers

Objective: Develop a compact wearable device that guides MCP and PIP flexion–extension while providing adjustable mechanical resistance.

Methodology: Anthropometric and subject-specific parameterisation of the finger; design of an RCM-based linkage generating virtual joint-centred rotation; iterative development of 3D-printed pneumatic chambers from early Flexojoint concepts to a pillow-shaped geometry; and integration of the pneumatic modules into a wearable glove and exoskeleton structure.

Validation: Quasi-static experimental characterisation in a sealed pneumatic circuit, measuring internal pressure, reaction force, and resistive torque over a bending range of 0° – 45° and for initial inflation pressures from 0.5 to 1.5 bar.

Main outcomes: The experiments demonstrate pressure-dependent and angle-dependent resistance. At 45° and an initial pressure of 1.5 bar, peak torques of approximately 0.42 N m and 0.30 N m are obtained for the forward and backward configurations, respectively. These results provide engineering evidence that the resistance level can be adjusted for different rehabilitation stages, while clinical effectiveness and closed-loop pressure regulation remain subjects for future work.

Chapter 4 – Model-Based Foundations for Static Feasibility and Design Optimization of Tendon-Driven Continuum Robots

Objective: Establish the modelling framework required to assess load-dependent static feasibility and to investigate the influence of tendon routing on feasible reach and mechanical effort.

Methodology: Discretised piecewise-constant-curvature kinematics and a recursive kineto-static formulation incorporating tendon tensions, gravity, vertebra-level friction, backbone elasticity, and external tip loads. Study A optimises the tendon-force distribution for a fixed robot geometry, whereas Study B performs a bi-objective dimensional synthesis of the tendon-routing radii and inter-segment angular offset.

Validation: Study A is evaluated through randomly sampled tendon-force vectors, optimisation convergence, and representative equilibrium configurations. Study B is evaluated through Pareto-front extraction for 100 randomised tendon-force realisations and statistical analysis of the Pareto-optimal solutions obtained across the optimisation runs.

Main outcomes: The chapter characterises the feasible static workspace under external loading and identifies practical routing trends, including a preference for larger angular offsets, particularly $\alpha \geq 30^\circ$, and tendon paths close to the outer admissible radii. The selected reference design, $r_1 = 21$ mm, $r_2 = 22.5$ mm, and $\alpha = 45^\circ$, is subsequently used in the planning studies of Chapter 5. The computational cost of repeated equilibrium calculations also motivates the transition toward learning-assisted and offline methods.

Chapter 5 – From Hybrid Task-Space Planning to Offline Capability-Map for Tendon-Driven Continuum Robots

Objective: Generate accurate and executable TDCR configurations while reducing the computational burden of repeated online inverse kinematics.

Methodology: First, supervised MLP and DNN models provide initial pose-to-configuration estimates that are refined through bound-constrained inverse kinematics. Second, an offline capability database is constructed by solving 54,000 voxel–orientation targets using PSO-based inverse kinematics and storing accepted configurations together with pose residuals, actuation commands, and Jacobian-based dexterity descriptors. Runtime selection is performed through Pareto filtering and normalised ideal-point ranking.

Validation: The hybrid method is evaluated over 16 trajectory–orientation combinations. In the reported Lock Y-axis comparison, the DNN validation RMSE decreases from 10.42 for the neural estimate to 5.85 after constrained refinement. The capability-map study evaluates solver acceptance, reachability, and dexterity distributions over a selected

region of interest containing 180 voxels, of which 88 are classified as reachable. A representative CAP-selected configuration is also executed on the physical prototype and measured using OptiTrack.

Main outcomes: Constrained refinement improves the accuracy of the neural initialisation but retains waypoint-wise online optimisation. The capability-map approach instead transfers inverse-kinematics and Jacobian computations to the offline stage, reducing runtime operation to database retrieval and ranking. The representative hardware study reports millimetre-range position errors, while also revealing larger orientation errors and the limitations of open-loop tendon actuation.

Chapter 6 – Prescribed-Time Fault-Tolerant Control of Tendon-Driven Continuum Robots

Objective: Develop safe tracking controllers with designer-specified convergence behaviour under nominal operation and under faults, delays, disturbances, and saturation.

Methodology: A reduced-order model of the dominant TDCR bending dynamics is used to formulate a nominal prescribed-time controller based on monotone tube boundaries, a transformed-error sliding structure, and an actor–critic learning component. The framework is then extended through a prescribed-time descriptor observer, an auxiliary saturation compensator, and a composite deep-learning-based fault-tolerant control law. Stability and boundedness properties are analysed using Lyapunov-based arguments.

Validation: Comparative numerical simulations are performed under sinusoidal, step–ramp, and parameter-variation conditions for the nominal controller and under step, sinusoidal, and multiple-fault scenarios for the fault-tolerant controller. The latter include actuator and sensor faults, time-varying input and output delays, disturbances, nonlinear uncertainty, and input saturation.

Main outcomes: In the nominal sinusoidal case, the reported effective settling time is reduced from approximately 0.75 s for the benchmark to approximately 0.2 s for the proposed controller, while the tracking error remains within the prescribed tube. In the fault-tolerant step-reference study, the effective settling time after reference changes is approximately 0.3 s, compared with approximately 5 s for the benchmark. The observer also provides rapid reconstruction of the robot states and output-channel corruption. These results establish numerical feasibility, while experimental implementation on the physical prototype remains future work.

Chapter 7 – Conclusions and Further Research

Objective: Integrate the findings of the rehabilitation and continuum-robot studies and identify the principal limitations and future research directions.

Methodology and validation basis: No new method or validation study is introduced; the chapter synthesises the mechanical, numerical, and experimental evidence presented in the preceding chapters.

Main outcomes: The chapter identifies the need for clinical evaluation and closed-loop pressure control of the rehabilitation system, broader experimental validation of the kineto-static and capability-map frameworks, trajectory-level capability-map extensions, closed-loop TDCR execution, and hardware implementation of the prescribed-time control strategies.

Taken together, the thesis progresses from application requirements and mechanism-level compatibility to load-dependent feasibility, constrained planning, capability-aware selection, and robust execution. This progression provides the common structure connecting the two application domains while preserving the distinct modelling and validation requirements of each.

Chapter 2

Biomechanical Foundations and State of the Art in Medical Robotics

Robotic systems for rehabilitation and minimally invasive interventions must be developed in a way that is consistent with human anatomy, biomechanics, and clinical practice. This chapter provides the background needed to contextualise the contributions of the thesis. Section 2.1 summarises hand rehabilitation, focusing on the biomechanical constraints that shape wearable device design and on representative technological approaches. Section 2.2 introduces tendon-driven continuum robots (TDCRs) and reviews the main modelling, design, and control challenges that arise in medical scenarios.

2.1 Hand Rehabilitation: Anatomy, Impairment, and Technology

The hand enables a large fraction of activities of daily living and is therefore central to functional independence [1]. Neurological and musculoskeletal impairments can reduce joint mobility, grasp strength, and coordination, often leading to long-term limitations [2]. These clinical needs motivate rehabilitation strategies that are intensive, repeatable, and adaptable across different patient capabilities [3]. These requirements are especially important for wearable devices intended to guide finger motion while remaining lightweight, compact, and comfortable for repeated use.

2.1.1 Anatomy and Biomechanics of the Finger

Wearable rehabilitation mechanisms must respect the kinematics of the finger joints and their interaction with soft tissues. In this thesis, the focus is on the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joints, which dominate functional flexion–extension for grasp and release. The MCP is a condyloid joint that supports flexion–extension and abduction–adduction, whereas the PIP behaves predominantly as a hinge joint in rehabilitation-oriented tasks. The distal interphalangeal (DIP) joint contributes to fingertip posture but is often secondary in device-level design decisions when the main objective is restoring basic grasping and release [4, 5].

A critical implication for exoskeleton design is that finger joint centres of rotation are not perfectly fixed during motion. Articular geometry and ligament constraints induce a migration of the instantaneous axis, so mechanisms that enforce a single fixed revolute axis can introduce misalignment, skin shear, and sliding at the human–device interface, especially over repeated training cycles. This motivates kinematic architectures that approximate a virtual joint centre without locating a physical joint exactly at the anatomical axis [6]. Remote Centre of Motion (RCM) mechanisms generate such a virtual centre of rotation while keeping the physical joints external, thereby improving kinematic compatibility in wearable devices.

2.1.2 Causes of Hand Impairment and Clinical Pathways

Hand impairments arise from conditions such as stroke, traumatic brain injury, spinal cord injury, peripheral nerve lesions, and orthopaedic trauma. Stroke, in particular, can lead to hemiparesis and altered muscle tone patterns that restrict independent movement of individual fingers. Rehabilitation practice commonly progresses from passive mobilisation, to preserve joint range and prevent contracture, toward active and active-assisted exercises, to promote motor re-learning, and finally to resistive training, to improve strength and endurance once voluntary motion becomes reliable [7]. From a device perspective, this means that the same system should support different interaction modes, ranging from low-resistance guidance to controllable resistance, while maintaining safe and repeatable contact forces.

2.1.3 Robot-Assisted Rehabilitation: Representative Systems and Design Trade-offs

Robotic rehabilitation technologies are often categorised into end-effector devices, rigid exoskeletons, and soft wearable gloves. End-effector devices offer general adaptability but

limited joint-specific guidance. Rigid exoskeletons can deliver higher forces and accurate motion guidance, but they may be bulky and sensitive to alignment [8]. Soft gloves provide lightweight and compliant interaction, yet they generally offer limited joint-level kinematic specificity and less direct regulation of clinically meaningful resistive torques [9].

Table 2.1 summarises representative wearable hand rehabilitation systems covering the main actuation families and design trade-offs, together with the hybrid RCM-pneumatic concept developed in this thesis.

This comparison shows that, for wearable hand rehabilitation devices, kinematic compatibility, adjustable resistance, and wearability must be considered together. In practical terms, this means that the mechanism should follow MCP/PIP flexion–extension while limiting relative motion between the device and the phalanges, provide adjustable resistance across different therapy stages, and remain compact, lightweight, and comfortable during repeated use.

2.2 Fundamentals of Tendon-Driven Continuum Robots

Minimally invasive procedures require long and slender instruments capable of navigating constrained anatomical pathways while interacting safely with delicate tissues. Continuum robots offer a promising solution through smooth, continuous deformation. Among them, tendon-driven continuum robots (TDCRs) are particularly attractive due to their remote actuation, compact distal structure, and compatibility with narrow anatomical corridors.

A TDCR typically consists of one or more compliant backbones with tendons routed along the robot body and anchored at the distal end. In vertebra-based designs, the backbone is guided through a sequence of vertebrae that define the tendon routing and segment geometry. Differential tendon tension generates bending and, depending on the structure, twisting. The robot behaviour depends strongly on structural stiffness, segment lengths, tendon routing geometry, and the distribution of vertebrae along the backbone, which together determine the robot's reachable workspace, load-bearing capability, and susceptibility to buckling or torsional effects. For medical applications, these design choices must therefore be considered together with tendon force limits and safety constraints [17].

2.2.1 Kinematic and Static Modelling

A common modelling baseline is the piecewise constant curvature (PCC) assumption, where each segment is approximated as a circular arc with constant curvature and a bending plane.

Table 2.1 Representative wearable hand rehabilitation systems and the proposed hybrid concept.

Picture	Device, actuation, and weight	Key features (pros)	Limitations (cons)
	Hand of Hope Rehab-Robotics [10] Rigid linkage; linear actuators ~700–800 g	• High-force, joint-guided rehabilitation • EMG-driven control for motor relearning • Precise kinematics	• Heavy and bulky on the hand • Non-portable desktop unit • Risk of joint misalignment
	Exo-Glove Poly In et al., 2015 [11] Soft/tendon routing; polymer-based 194 g	• Very compact design • Separation of glove and actuator • Soft tendon routing	• Risk of hyperextension • Friction in tendon paths • Actuates multiple joints together
	Harvard Soft Glove Polygerinos et al. [12] Soft/pneumatic; fiber-reinforced ~285 g	• Fluid-driven hydraulic actuation • High bending capability • Portable belt pack	• Low blocking force, approximately 8 N • Slow response time • Bulky fluid reservoir
	PneuNets Glove Yap et al., 2015 [13] Soft/pneumatic; silicone elastomer ~200 g	• Rapid bending response • Simple fabrication • Inherently safe	• Ballooning-effect risks • Low force output, below 10 N • Material fatigue over time
	High-Force Fabric Glove Ge et al., 2020 [14] Soft/pneumatic; TPU-coated fabric 128 g	• High blocking force, approximately 38 N • Durable heat-sealed structure • Very lightweight	• Reduced flexibility • Bulky dorsal volume • Reduced tactile sensation
	Fabric Soft Glove Cappello et al. [15] Soft/pneumatic; textile actuators 77 g	• Ultralight design • High user acceptance • Suitable for light ADL support	• Low force output, approximately 15 N • Leakage issues in prototypes • Limited to light ADLs
	HPA Soft Glove Lai et al., 2023 [16] Soft/pneumatic; honeycomb fabric actuator 107 g with glove; 1.8 kg control box	• High force output, approximately 13.6 N tip force • Bidirectional guidance for active flexion and extension • Efficient actuation at 150 kPa	• Open-loop control • Lacks rigid guidance for misalignment prevention
	Proposed System This thesis [6] Hybrid; RCM guidance + inflatable chambers ~200 g	• Joint-level kinematic compatibility • Pressure-tunable resistance • Wearable architecture	• Requires custom mechanical integration and fabrication • Lower bulk than many rigid exoskeletons, but higher than textile-only gloves

PCC enables efficient forward kinematics and is useful for large-scale sampling and dataset generation. However, classical PCC formulations may become inaccurate when torsion and external loads play a significant role. For this reason, extended models incorporate axial twist, providing improved geometric fidelity for multi-segment robots.

In addition to kinematics, kineto-static modelling is essential whenever the robot must hold configurations under external forces and torques. These formulations couple geometry with static equilibrium, linking tendon tensions, backbone elasticity, and external loads. Practical deviations from model predictions commonly arise from friction, backlash, and manufacturing tolerances, motivating more robust design choices as well as compensation strategies at the control level [18].

2.2.2 Workspace Analysis: Feasible Static Workspace and Capability Maps

Geometric reachability alone is not sufficient in medical applications. A configuration is useful only if it can be maintained in static equilibrium under admissible tendon forces and clinically relevant loading conditions. This motivates the concept of the Feasible Static Workspace (FSW), defined as the subset of the geometric workspace that satisfies both static equilibrium and actuation constraints. In practice, evaluating the FSW is challenging because tendon forces, backbone elasticity, and external loads are strongly coupled [18].

Capability maps extend this analysis by associating additional information with feasible task-space regions. Depending on the application, these descriptors may include orientation, dexterity, stiffness, or energy-related measures. Such maps help identify regions where the robot can operate more effectively under constraints, and they provide a compact link between workspace geometry and functional performance.

2.2.3 Advanced Control and Learning-Based Approaches

Accurate operation of TDCRs requires control strategies that can cope with nonlinear behaviour, model uncertainty, and changing operating conditions. Purely kinematic control can be simple to implement, but its performance often degrades when external loads, friction, or modelling errors become significant. For this reason, model-based approaches have been developed to incorporate static or dynamic effects more explicitly into the control design.

At the same time, the complexity of continuum-robot models has encouraged the use of learning-based and hybrid methods. Supervised learning can be used to approximate

inverse or forward mappings from data, while hybrid schemes combine data-driven estimation with optimisation or model-based correction. In addition, safety-critical applications have motivated the study of prescribed-time and fault-tolerant control strategies, especially in the presence of delays, saturation, disturbances, or sensor and actuator faults. These developments highlight the need for control methods that remain accurate, robust, and computationally manageable under practical operating conditions.

In this thesis, these developments motivate two complementary directions: feasibility-aware planning through hybrid and database-driven methods, and robust control through prescribed-time and fault-tolerant formulations.

2.3 Chapter Summary

This chapter established the biomechanical and technological context for the compliant robotic systems developed in this thesis. For hand rehabilitation, it highlighted why MCP/PIP biomechanics and migrating centres of rotation impose stringent constraints on wearable mechanisms, and it summarised how existing technologies trade kinematic precision against compliance and wearability. These considerations motivate the hybrid solution pursued here, combining RCM-based kinematic compatibility with pressure-tunable resistance.

For continuum robotics, the chapter introduced TDCR fundamentals and summarised the modelling, workspace, and control challenges that arise in medical environments. The distinction between geometric workspace and feasible static workspace, together with capability mapping, provides the basis for the design optimisation and analysis frameworks developed in the following chapters.

Part I

Section Two

Chapter 3

Hand Rehabilitation Systems: RCM Exoskeleton and Inflatable Chambers

3.1 Anatomical Basis and Geometric Parameters of the Finger

This chapter focuses on the rehabilitation of a long finger, using the index finger as the representative case for mechanism design and experimental characterization. Each long finger consists of the proximal, middle, and distal phalanges, connected by the metacarpophalangeal (MCP), proximal interphalangeal (PIP), and distal interphalangeal (DIP) joints, as shown in Figure 3.1. In the present work, the main focus is on the MCP and PIP joints, as they govern the flexion–extension motion that is most relevant for finger opening and closing during rehabilitation tasks.

For the device developed in this chapter, the anatomical description is limited to the geometric features required for mechanism scaling and placement. In particular, phalanx lengths are required to define the dimensions of the wearable structure and to initialize the CAD layout of the RCM-based mechanism.

To parameterise the wearable mechanism, phalanx lengths are reported from two sources. The first is an anthropometric reference dataset, used to provide representative long-finger dimensions across subjects. The second is a set of subject-specific measurements collected from the author’s hand and used for the prototype presented in this thesis. In this way, the reference values provide a general basis for design, while the measured values correspond to the actual geometry used for prototype development.

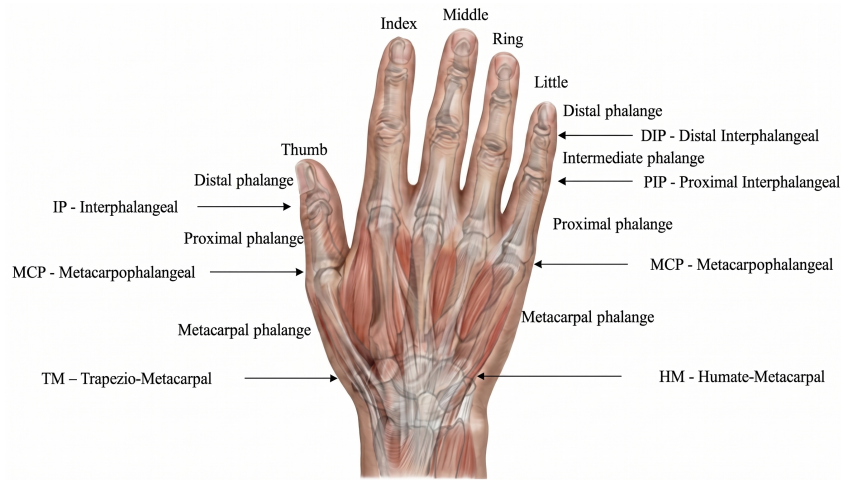


Figure 3.1 Anatomical details of the hand skeleton.

The reference dataset used in this study is the long-finger phalanx length table reported by Habib and Kamal [19], which provides distal, middle, and proximal phalanx lengths for the index, middle, ring, and little fingers. For clarity in design parameterisation, Table 3.1 reports the overall mean values obtained by averaging the data for both sexes and both hand sides.

Table 3.1 Mean long-finger phalanx lengths in millimetres, averaged across sex and side [19].

Finger	Distal	Middle	Proximal
Index	22.7	23.1	25.2
Middle	24.7	26.4	26.6
Ring	22.1	24.6	26.4
Little	18.7	18.5	23.8

The prototype geometry was then defined using subject-specific measurements collected from the author's hand. These values are reported in Table 3.2 and are used in the following sections to scale the RCM linkage and initialize the CAD model.

Table 3.2 Subject-specific phalanx lengths used for prototype parameterisation in millimetres.

Finger	Distal	Middle	Proximal
Index	24.0	27.0	26.7
Middle	26.7	28.4	27.6
Ring	23.2	25.7	27.2
Little	18.7	18.5	23.8

3.2 Problem Statement and Design Objectives

The background in Chapter 2 showed that wearable hand rehabilitation devices must satisfy two main requirements. First, the mechanism should remain kinematically compatible with the anatomical joints, so that flexion–extension can be guided while limiting joint misalignment and relative motion at the finger–device interface. Second, the device should provide adjustable resistance across different therapy stages, from passive movement to active and resistive training, while remaining compact and comfortable for repeated use.

These requirements are not easily satisfied by a single conventional solution. Rigid exoskeletons can provide accurate motion guidance, but their performance depends strongly on alignment and mechanical integration. Soft gloves offer compliant interaction and reduced weight, but they provide less direct joint-level guidance and less repeatable resistance during structured training. For this reason, the system developed in this chapter follows a hybrid approach.

The proposed design combines two functional layers. The first is an RCM-based exoskeleton that guides MCP and PIP flexion–extension through virtual centres located close to the anatomical joint regions. The second is a pneumatic resistance layer based on inflatable chambers integrated into a wearable glove. As the finger bends, the chamber is compressed and generates a pressure-dependent resistive response.

The design objective is therefore to obtain rehabilitation-relevant MCP/PIP motion with acceptable kinematic compatibility, while also providing adjustable resistance in a wearable and compact form. In practical terms, the device should remain low-profile, lightweight, and easy to wear, while producing a torque–angle response that can be adjusted through the initial chamber pressure. The overall concept of the proposed system, including the RCM linkage and the pneumatic chamber arrangement, is shown in Figure 3.2.

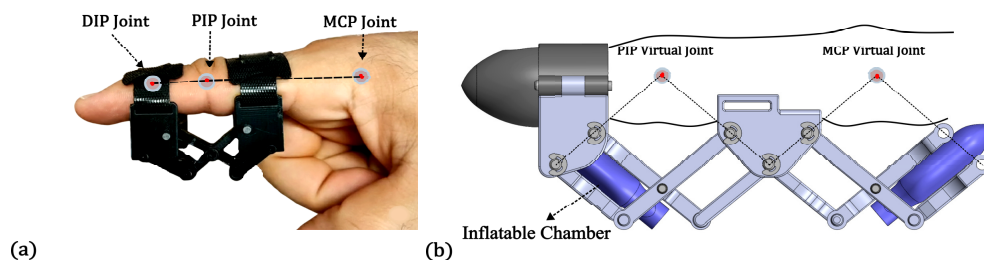


Figure 3.2 Concept overview of the hybrid rehabilitation system. (a) Index finger anatomy and joint locations (DIP, PIP, MCP). (b) CAD model illustrating the RCM-based linkage and the placement of inflatable chambers.

3.3 Design Concept and Kinematic Principle of the RCM Exoskeleton

To improve kinematic compatibility with finger flexion–extension, a compact exoskeleton architecture was developed based on the Remote Centre of Motion (RCM) principle. The mechanism is designed to create a virtual centre of rotation within the anatomical joint region, while the physical pivots remain outside the joint. In this way, the flexion motion can be guided around a joint-centred virtual axis without placing a mechanical hinge directly at the anatomical joint.

The mechanism is implemented as a compact four-link arrangement, illustrated here for the PIP joint in Figure 3.3. The linkage geometry follows the natural curvature of the finger and preserves a low-profile configuration during motion. Its dimensions are defined using the phalanx length data introduced in Section 3.1, allowing the mechanism to be scaled to different finger sizes while remaining suitable for wearable integration.

In the rest posture, the mechanism is in its open configuration, and the movable forward part is at its maximum separation from the fixed backward part. This initial spacing is denoted by B , as shown in Figure 3.3. As the finger flexes, the RCM linkage closes while maintaining the virtual joint-centred guidance. At the same time, the spacing between the two parts decreases from B to a smaller distance d . Since the inflatable chamber is positioned between these two parts, this reduction directly compresses the chamber.

Before bending starts, the chamber is inflated to a predefined initial pressure, and the pneumatic circuit is isolated by closing the hand-operated valve, so that the chamber and the connecting hoses form a sealed air volume, as shown in Figure 3.3. Under these conditions, chamber compression increases the internal pressure and generates a reaction force that opposes the motion, thereby producing a resistive torque about the joint. As flexion progresses, d decreases further and the resistive response increases accordingly. Figure 3.3 illustrates the geometric layout of the mechanism and its coupling with the pneumatic chamber.

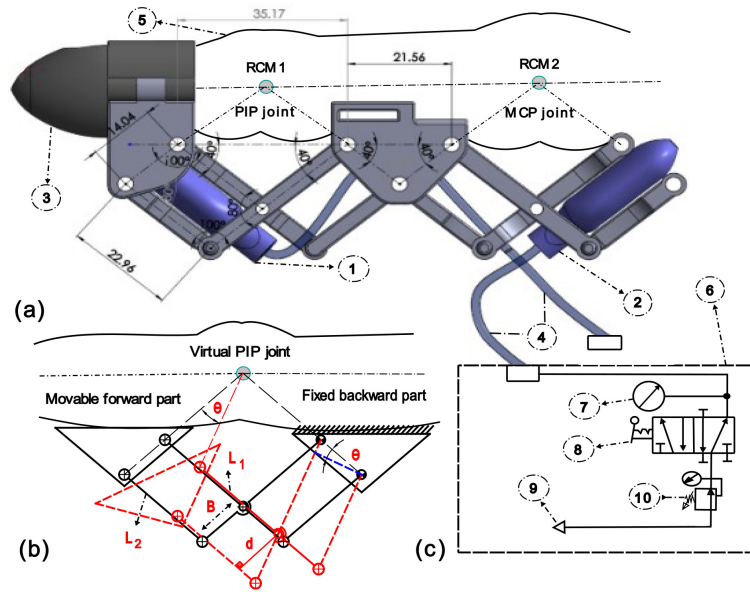


Figure 3.3 RCM-based finger mechanism and its coupling with the sealed pneumatic chamber system used to generate pressure-dependent resistance during flexion. (a) Side view with dimensions in millimetres. (b) Rotation mechanism. Numbers correspond to: 1 = forward chamber, 2 = backward chamber, 3 = thimble, 4 = air tube, 5 = index finger, 6 = pneumatic system, 7 = pressure sensor, 8 = hand lever valve, 9 = air supply, 10 = air regulator.

3.4 Development of Compliant Variable-Stiffness Modules

While the RCM linkage provides the kinematic guidance layer, rehabilitation also requires an adjustable resistive element that can be adapted to different therapy stages. For this reason, inflatable chambers are integrated into the wearable structure and mechanically coupled to the guided finger motion. The development of these chambers did not proceed directly to the final design. Instead, several prototype generations were examined in order to improve contact stability, repeatability, and sealing.

3.4.1 From Flexojoint to Pillow-Shaped Chambers

The early prototypes were based on barrel-type “Flexojoint” chambers. These first designs showed that pressurised deformation could be used to generate resistance during bending. However, the curved external profile led to unstable contact and limited repeatability under repeated inflation and compression. Subsequent bellows-type geometries improved deformability, but they remained sensitive to non-uniform folding and local instabilities, which affected both repeatability and sealing. The final selection was therefore guided by three

practical criteria: contact stability during compression, repeatability of the force response, and reliable sealing under repeated inflation cycles.

For these reasons, the final design adopted in this thesis is a compact pillow-shaped chamber placed between rigid plates. This geometry provides a flatter contact profile and a more predictable compression behaviour during bending. Fabrication is carried out by fused deposition modelling (FDM): the inflatable chamber walls are printed in TPU, whereas the rigid mounting and supporting parts are printed using PLA-based materials. The wall thickness is tuned iteratively to balance deformability and sealing, and the prototypes reported in this chapter use a thickness of 0.9 mm. A soft TPU fingertip thimble is also introduced for distal attachment in order to improve comfort and interface quality. The main stages of chamber development are shown in Figure 3.4.

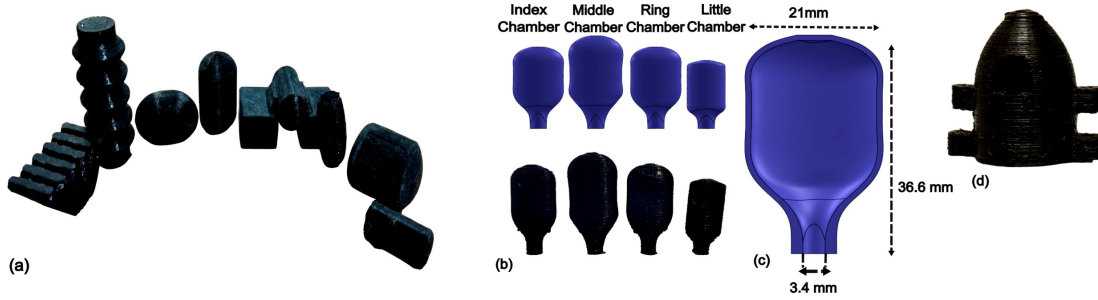


Figure 3.4 Chamber design evolution and final geometry. (a) Early Flexojoint prototype. (b) Intermediate concept. (c) Final pillow-shaped chamber with rigid plates and TPU inflatable cavity. (d) Distal thimble integration.

3.4.2 Glove Integration and Coupling to the RCM Layer

To extend the variable-resistance concept from an isolated module to a wearable rehabilitation device, the pillow-shaped chambers are integrated into a glove structure positioned near the MCP and PIP regions. The glove provides both a mounting layer and a compliant interface with the finger, while the RCM exoskeleton is assembled over it to guide the flexion motion.

In the integrated design, the chamber is placed so that its compression is mechanically coupled to the closure of the RCM-guided motion described in Section 3.3. Different resistance levels are obtained by setting an initial inflation pressure, p_{init} , before bending. In the quasi-static experiments reported later in this chapter, the pneumatic circuit is isolated by closing a manual valve. The integration of the chambers into the glove and the assembly of the exoskeleton are shown in Figure 3.5.

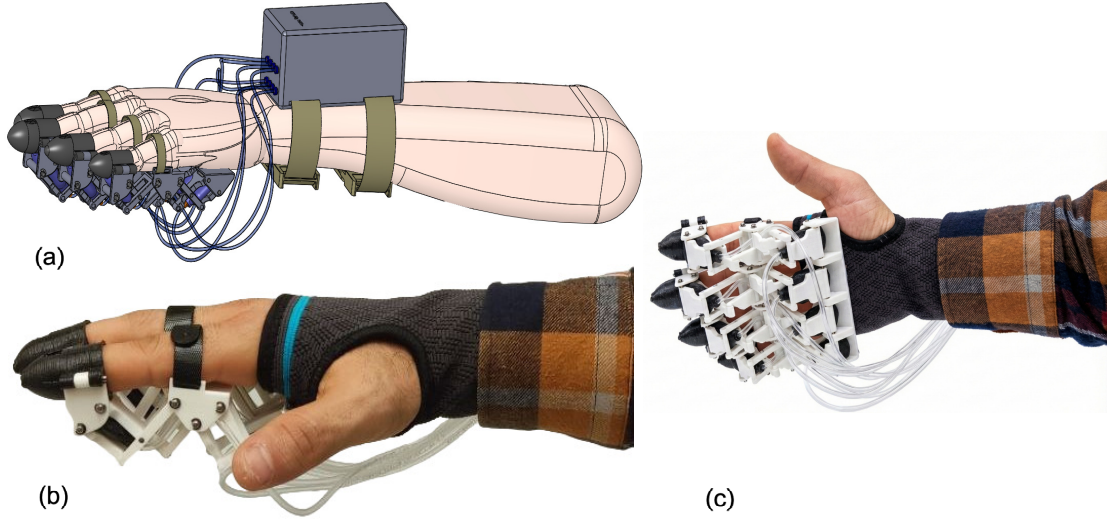


Figure 3.5 System integration. (a) CAD layout showing chamber placement near MCP and PIP. (b) Assembly on the hand with the exoskeleton mounted over the glove. (c) Prototype with pneumatic lines connected.

3.5 Experimental Characterization

This experimental study quantifies the relationship between bending angle θ , internal pressure p , reaction force F , and resistive torque τ for the final pillow-shaped chambers, both as isolated modules and in the glove-integrated configuration.

3.5.1 Experimental Setup and Procedure

The experimental setup consists of a mechanical fixture, a closed pneumatic circuit, and measurement instrumentation. The fixture includes a fixed base and a rotating arm representing the bending link, with the bending angle measured by a goniometer. The pneumatic circuit includes a regulated air supply, hoses, and a manual valve. Internal gauge pressure is measured using an MPX2200GP sensor from 0 to 200 kPa, while the reaction force is measured using a digital dynamometer applied at a known lever arm S . Data are acquired using a Teensy 3.5 board.

For each test, the chamber is inflated to one of three prescribed initial pressures: 0.5, 1.0, or 1.5 bar. The valve is then closed to isolate the pneumatic circuit. The structure is subsequently bent quasi-statically from 0° to 45° . At discrete angular steps, the pressure $p(\theta, p_{\text{init}})$ and the reaction force $F(\theta, p_{\text{init}})$ are recorded. The resistive torque is computed as

$$\tau(\theta, p_{\text{init}}) = F(\theta, p_{\text{init}})S, \quad (3.1)$$

where S is the moment arm of the force application point. In the present setup, $S_1 \approx S_{PIP} \approx 35$ mm, and $S_2 \approx S_{MCP} \approx 28$ mm. Two chamber configurations are evaluated, referred to as forward and backward. The experimental setup used for chamber characterisation is shown in Figure 3.6.

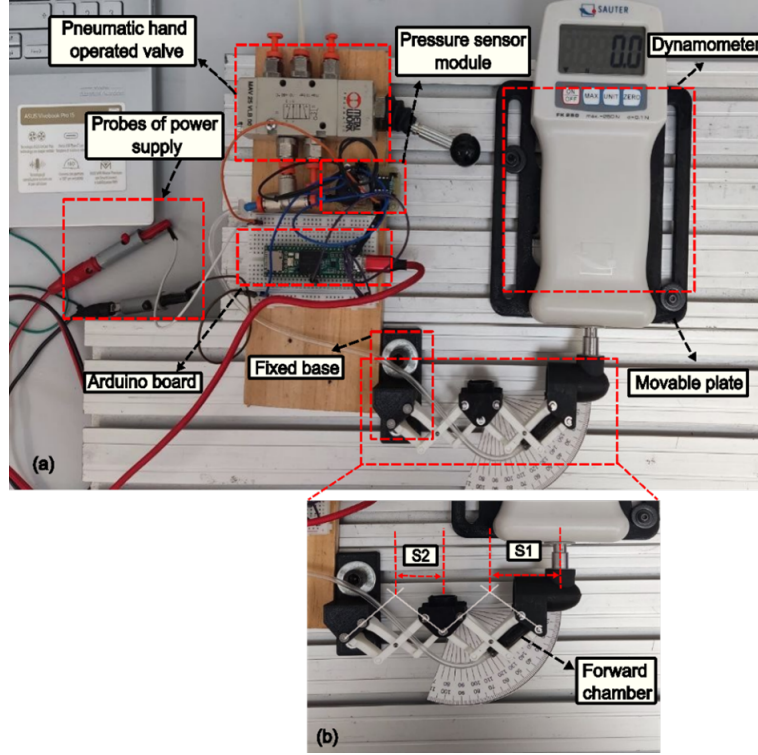


Figure 3.6 Experimental setup used to characterise the pillow-shaped chambers, including the bending fixture, sealed pneumatic circuit, pressure sensor, manual valve, and dynamometer.

3.5.2 Pressure Evolution

The internal pressure increases with bending angle, and the magnitude of the increase depends on p_{init} . At small bending angles, approximately below 10° , pressure changes remain limited, which is consistent with modest chamber compression. As bending increases, compression becomes more pronounced and the pressure rise becomes clearer under sealed conditions. The forward and backward configurations show different pressure slopes, which is consistent with their different enclosed air volumes. Under $p_{init} = 1.5$ bar, the forward chamber reached a maximum of approximately 1.75 bar at 45° , whereas the backward chamber reached approximately 2.0 bar under the same conditions. The evolution of the measured pressure for different initial pressures is shown in Figure 3.7.

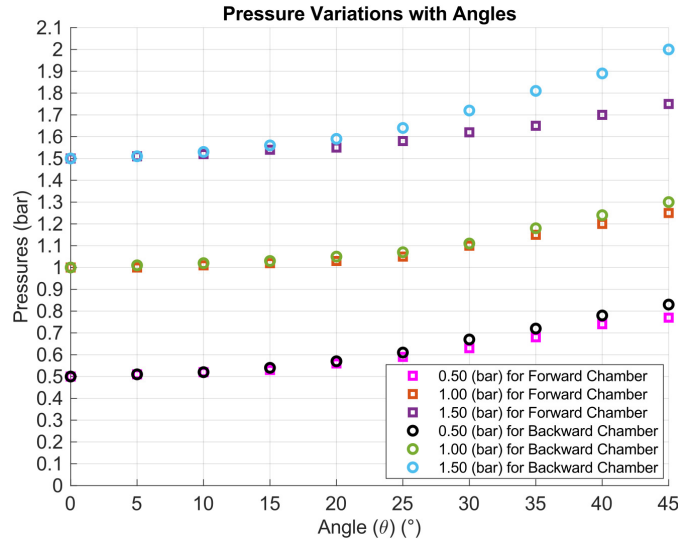


Figure 3.7 Pressure evolution $p(\theta, p_{\text{init}})$ for forward and backward chambers for different initial pressures.

3.5.3 Force Evolution

The bending force $F(\theta, p_{\text{init}})$ increases with both bending angle and initial pressure. For a fixed bending angle, a larger p_{init} produces a larger reaction force. The forward and backward configurations also show different force levels, consistent with the pressure behaviour described in the previous subsection. At the maximum tested bending angle of 45° , the force reached approximately 9.2 N in the forward chamber and 9.5 N in the backward chamber for $p_{\text{init}} = 1.5$ bar, whereas for $p_{\text{init}} = 0.5$ bar the corresponding values were only about 5.2–5.5 N. The measured bending-force curves for different initial pressures are shown in Figure 3.8. These results confirm that the chamber response can be modulated both by bending and by the selected initial pressure.

3.5.4 Torque Response

Using the measured force and corresponding lever arm, the resistive torque $\tau(\theta, p_{\text{init}})$ is computed for MCP and PIP configurations. The resulting torque–angle curves confirm that the resistive response can be adjusted through p_{init} , allowing the same chamber geometry to produce different torque profiles over the flexion range. The response is nonlinear: the effective stiffness, $d\tau/d\theta$, is lower near extension and increases at larger flexion angles, indicating progressive stiffening as compression increases.

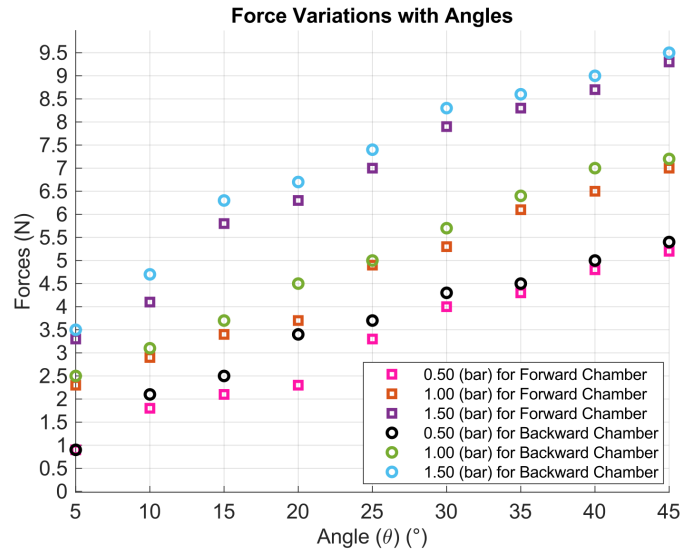


Figure 3.8 Measured bending force in the forward and backward chambers across bending angles for different initial pressures.

For the tested conditions, the forward configuration generated up to approximately 0.42 Nm, while the backward configuration produced about 0.30 Nm at 45° and $p_{\text{init}} = 1.5$ bar. These values indicate that the actuator can provide measurable and adjustable resistance for guided motion and light resistive training. More generally, the results show that the torque output can be modulated by adjusting the initial inflation pressure. The corresponding torque–angle curves are shown in Figure 3.9.

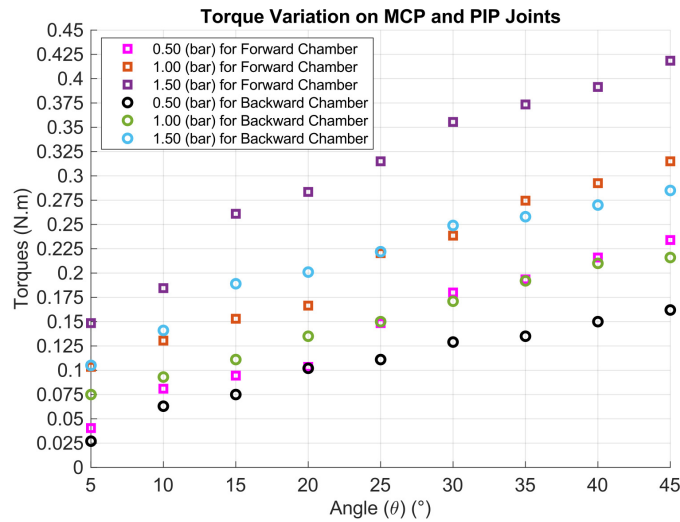


Figure 3.9 Resistive torque $\tau(\theta, p_{\text{init}})$ at MCP and PIP joints for different initial pressures.

3.6 Rehabilitation Implications and Limitations

The experiments show that the device provides a controllable relationship between joint angle, internal pressure, force, and resistive torque. In practical terms, this means that the resistive response can remain relatively low at small bending angles and increase at larger flexion angles, while its overall magnitude can be adjusted through p_{init} . This makes the same device suitable for different rehabilitation modalities, including passive mobilisation at lower pressure levels and resistive training at higher pressure levels, provided that safe limits on pressure and interface loading are respected.

At the same time, the present characterization remains quasi-static and focuses on repeatable torque–angle behaviour under controlled bending conditions. The wearable prototype is presented here as an engineering validation of the coupled RCM pneumatic concept. Clinical evaluation across different patient groups, as well as real-time closed-loop pressure control, are beyond the scope of this chapter and are left for future work.

3.7 Summary

This chapter presented a hybrid wearable system for hand rehabilitation combining an RCM-based finger exoskeleton with glove-integrated inflatable chambers. The mechanism provides guided MCP/PIP motion through virtual joint-centred rotation, while the pneumatic layer introduces adjustable resistance through chamber compression during bending.

The chamber design was refined through iterative prototyping toward a pillow-shaped geometry suitable for wearable integration. An experimental study then characterised the relationship between bending angle, pressure, force, and resistive torque under sealed pneumatic conditions. Together, these results establish the mechanical basis of the proposed rehabilitation device and show that the resistive response can be adjusted through the initial inflation pressure.

With the rehabilitation system established and characterised, the thesis now moves to its second part. Chapter 4 carries this constraint-aware perspective into tendon-driven continuum robots, replacing purely geometric reachability with an analysis of static feasibility under actuation, structural, and external-loading constraints, and using this framework for workspace analysis and tendon-routing dimensional synthesis.

Chapter 4

Model-Based Foundations for Static Feasibility and Design Optimization of Tendon-Driven Continuum Robots

4.1 Motivation and Scope of the Chapter

The previous chapter showed that the behaviour of a compliant rehabilitation device cannot be understood from geometry alone. In the proposed hand rehabilitation system, the mechanical response was shown to depend on deformation, internal pressure, reaction force, and resistive torque. A similar principle applies to tendon-driven continuum robots (TDCRs). For these robots, geometric reachability alone does not determine practical feasibility, since admissible configurations must also satisfy actuation limits, structural elasticity, friction, gravity, and external loading. In the second part of this thesis, these effects are addressed through a model-based study of static feasibility and tendon-routing design.

TDCRs are attractive for applications that require slender, compliant robots capable of operating in confined or delicate environments, such as minimally invasive interventions, endoluminal procedures, and industrial inspection [20–22]. At the same time, their continuous deformation, actuation redundancy, and load-dependent behaviour make it difficult to identify configurations that are not only geometrically reachable but also maintainable in static equilibrium with acceptable mechanical effort. For this reason, a purely kinematic description is not sufficient for the design questions considered in this thesis.

Most kinematic descriptions of TDCRs rely on the piecewise constant curvature (PCC) assumption, which provides an efficient representation of backbone geometry and enables

tractable forward kinematics [23–25]. Extended kinematic and dynamic descriptions have also been proposed using Clarke-transform-based control, dual-quaternion kinematics, and rod-based formulations [26–28]. However, when tendon loading, gravity, friction, backbone elasticity, and external tip forces must be considered together, a kineto-static formulation becomes necessary [29–31]. More detailed static and dynamic modelling studies have further considered experimentally evaluated cable-driven structures, Cosserat-rod formulations, continuum parallel robots, variable-curvature continuum joints, and strain-based sensing for shape and force estimation [32–36]. Within this setting, workspace analysis naturally shifts from geometric reachability to feasible static workspace (FSW), namely the set of tip positions that can be maintained in static equilibrium under tendon-force bounds and external loading [37–41]. The same framework also makes it possible to study how tendon-routing geometry influences both feasible reach and actuation efficiency [42–46].

Against this background, this chapter establishes the model-based foundation for the remainder of the TDCR part of the thesis. First, a PCC-based kineto-static formulation is developed for a two-segment vertebra-based TDCR under tendon forces and external loads. Second, this formulation is used to characterise the feasible static workspace for a fixed robot design through tendon-force optimisation. Third, the same framework is extended to a bi-objective dimensional synthesis problem in which the tendon-routing radii and the inter-segment angular offset are optimised to balance feasible static reach and mechanical work.

The chapter begins with the robot architecture and the modelling assumptions used in the analyses that follow. A discretised PCC representation is then introduced, followed by a kineto-static formulation that accounts for tendon forces, gravity, vertebra-level friction, backbone elasticity, and external tip loads. On this basis, two optimisation studies are carried out. Study A considers a fixed robot geometry and optimises tendon-force distributions to characterise feasible static reach under load. Study B treats tendon-routing parameters as design variables and extracts practical design guidelines from a bi-objective optimisation and statistical analysis of the obtained solutions.

The results presented here provide the model-based foundation for the chapters that follow. The selected routing design and the kineto-static framework support the capability-map and hybrid inverse-kinematics methods developed in Chapter 5, while the same robot architecture is used in Chapter 6 for prescribed-time and fault-tolerant control under practical constraints. In this way, the present chapter links the compliant-system perspective introduced in the first part of the thesis to the planning and control methods developed later for tendon-driven continuum robots.

4.2 Robot Architecture and Modelling Assumptions

The continuum robot considered in this part of the thesis is a two-segment tendon-driven continuum robot (TDCR) with a vertebra-based structure. Each segment consists of a flexible backbone and a sequence of vertebrae that guide the tendons and define the spatial discretisation used in the model. The same overall architecture is used throughout the chapter, with two closely related configurations. The first is a baseline configuration used for feasible static workspace analysis in Study A. The second is a prototype-matched configuration used for the dimensional synthesis study in Study B. Their main geometric and mechanical parameters are reported in Table 4.1.

4.2.1 Mechanical Structure and Tendon Routing

Both studies use the same basic layout: two serially connected segments, each actuated by four tendons. The tendons pass through guide holes in every vertebra and are arranged as two antagonistic pairs. In each segment, the tendon channels are placed at fixed angular positions around the backbone centreline, so that differential tension within one pair generates bending in the corresponding plane, while coordinated actuation of both pairs enables general spatial bending.

In Segment 1, the tendon channels are aligned with the local cross-sectional axes at angular positions 0° , 90° , 180° , and 270° . In Segment 2, the same pattern is rotated by an inter-segment angular offset α . In the baseline configuration adopted in Study A, this offset is fixed at $\alpha = 45^\circ$. In Study B, by contrast, α is treated as a design variable in order to evaluate its influence on feasible static workspace and actuation effort.

The actuation unit is based on pulley-driven tendons operated by Dynamixel 2XL430-W250-T dual-axis servomotors. Each segment is controlled by two antagonistic tendon pairs positioned 180° apart. Bending is generated by increasing the tension in one tendon while releasing its opposite counterpart. The tendons are made of 0.5 mm-diameter fishing line, and the continuum body is fabricated as a monolithic PA12 structure using selective laser sintering (SLS). Figure 4.1 shows the actuation unit and the tendon-drive arrangement.

Although both studies follow the same architectural template, they do not use identical numerical parameters. Study A adopts a baseline configuration for actuation-level feasible workspace analysis, whereas Study B uses the prototype-matched configuration employed for design optimisation and for the subsequent chapters. Table 4.1 therefore reports the corresponding parameters separately.

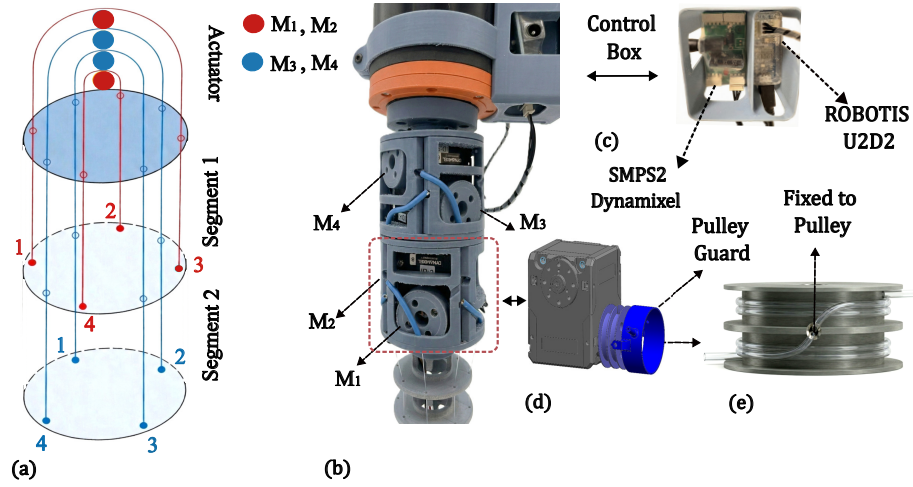


Figure 4.1 Diagram of the actuator unit and drive mechanism. (a) Driving cable arrangement, with tendons 1–4 for Segment 1 shown in red and tendons 1–4 for Segment 2 shown in blue. (b) Actuator unit with motors M1–M4 arranged with a 45° offset. (c) Control unit. (d) Motor with mounted pulley and protective guard. (e) Pulley mechanism detail.

Table 4.1 Geometric and mechanical parameters of the two-segment TDCR.

Parameter	Study A	Study B	Unit	Notes
Number of vertebrae per segment (N)	10	7	–	–
Segment length (L)	0.2	0.13263	m	–
Tendon routing radius (r_m)	5 (fixed)	varied	mm	–
Vertebrae radius (r_d)	6	24.7	mm	–
Backbone outer radius (r_o)	1.4	12.56	mm	–
Backbone inner radius (r_i)	0	9.56	mm	–
Tendon channels (d_1, d_2)	1.5	1.5	mm	–
Young's modulus (E)	54	54	GPa	–
Poisson's ratio (ν)	0.3	0.3	–	–
Vertebrae mass (m_d)	9.2	9.2	g	Per vertebra
Backbone mass per unit length (m_b)	0.0115	0.0115	kg/m	–

External loading is introduced at the distal tip in the form of a force vector $\mathbf{F}_{\text{ext}}$ and, when required, a moment vector $\mathbf{M}_{\text{ext}}$. These loads are propagated backward through the discretised structure in the kineto-static solver.

4.2.2 Modelling Assumptions

The kinematic and kineto-static analyses in this chapter are based on a set of simplifying assumptions chosen to retain the main physical effects while keeping the model computationally tractable for repeated optimisation.

First, each segment is discretised into backbone subsegments between consecutive vertebrae, and each subsegment is represented using the piecewise constant curvature (PCC) assumption. This means that the local backbone shape within each subsegment is approximated as a circular arc with constant curvature and, where required, local twist. This representation is consistent with the vertebra-based architecture and provides a compact parametrisation for both forward kinematics and equilibrium analysis.

Second, the chapter is restricted to static or quasi-static conditions. Inertial effects are neglected, and robot shapes are computed from force–moment equilibrium under tendon forces, gravity, friction, backbone elasticity, and externally applied tip loads. Dynamic effects are therefore outside the scope of this chapter and are addressed later only in the context of control.

Third, the tendons are assumed to be massless, inextensible, and capable of transmitting tensile forces only. Friction at the tendon guides is represented by a vertebra-level Coulomb model based on the local tendon deflection angle. This approximation captures the main tension losses along the robot without introducing the additional complexity of a full distributed friction model.

Fourth, the backbone is treated as linearly elastic in bending and, when included, in torsion. The vertebrae are assumed to be rigid relative to the compliant backbone. This allows the restoring effect of the backbone to be described through elastic bending and torsional moments, while the vertebrae act primarily as tendon-routing and discretisation elements.

These assumptions define the modelling framework used throughout the remainder of the chapter. On this basis, the next section formulates the discretised PCC kinematic model, which is then extended to the kineto-static equilibrium analysis used in Studies A and B. Figure 4.2 summarises the overall robot structure and the tendon-routing geometry.

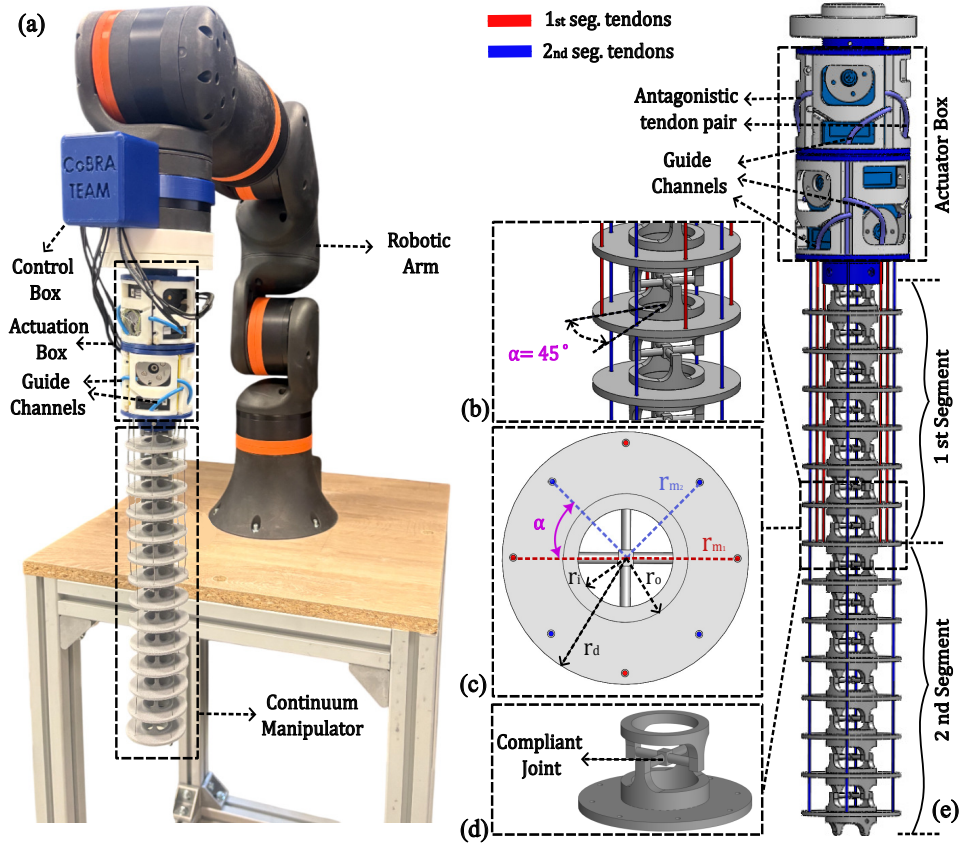


Figure 4.2 Overview of the two-segment tendon-driven continuum robot prototype and its structural layout. (a) Experimental setup with actuation unit and control box. (b) Four-tendon arrangement on each segment and inter-segment angular offset α . (c) Tendon-routing geometry. (d) Vertebra and compliant joint. (e) CAD model of the proposed structure.

4.3 Kinematic Modelling of TDCR

The kinematic model describes how the backbone shape and tip pose of the two-segment tendon-driven continuum robot depend on the curvature parameters of its subsegments. A piecewise constant curvature (PCC) representation is adopted, consistent with the vertebra-based architecture introduced in Section 4.2. This section presents the discretised PCC parametrisation and the homogeneous transformations between consecutive vertebra frames.

4.3.1 Discretized PCC Representation

Each segment is discretised into subsegments by the vertebrae, and the backbone portion between two consecutive vertebrae is modelled as a constant-curvature subsegment of arc length δ_i . For subsegment i , the curvature components (β_i, γ_i) define the curvature magnitude

κ_i and the bending plane orientation φ_i as:

$$\kappa_i = \sqrt{\beta_i^2 + \gamma_i^2}, \quad \varphi_i = \text{atan2}(\gamma_i, \beta_i) \quad (4.1)$$

The bending angle and twist angle over subsegment i are then defined as

$$\theta_i = \kappa_i \delta_i, \quad \rho_i = \tau_i \delta_i \quad (4.2)$$

where τ_i represents the local twist rate about the backbone axis. The PCC state of subsegment i is therefore described by $(\beta_i, \gamma_i, \tau_i)$, from which κ_i , φ_i , θ_i and ρ_i are obtained. This parametrization is used both in the forward kinematic chaining and in the kineto-static equilibrium model developed later in this chapter. Figure 4.3 shows the PCC parametrization of a single subsegment and its relation to consecutive vertebra frames.

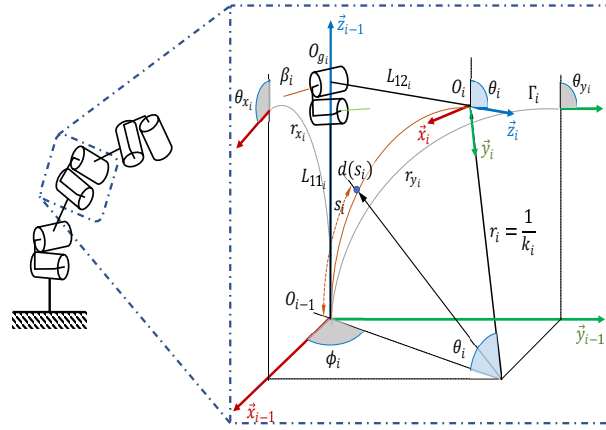


Figure 4.3 PCC parametrisation of a single backbone subsegment between two consecutive vertebra frames.

4.3.2 Subsegment Transformation and Forward Kinematics

Let $(i-1)$ and (i) denote the local frames attached to vertebrae $(i-1)$ and i , respectively, with their z -axes aligned with the local backbone direction. Under the PCC assumption, the endpoint position of subsegment i , expressed in frame $(i-1)$, is

$$\mathbf{p}_i = \begin{bmatrix} \frac{1 - \cos \theta_i}{\kappa_i} \\ 0 \\ \frac{\sin \theta_i}{\kappa_i} \end{bmatrix} \quad (4.3)$$

The corresponding rotation can be expressed as a sequence of elementary rotations: a rotation about the z -axis that aligns the reference plane with the bending plane, a rotation about the y -axis by the bending angle, and a rotation about the z -axis that accounts for

twist. Using φ_i for the bending plane angle and ρ_i for the twist angle, the homogeneous transformation from frame $(i-1)$ to frame (i) is written as:

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_z(\varphi_i)\mathbf{R}_y(\theta_i)\mathbf{R}_z(\rho_i - \varphi_i) & \mathbf{p}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (4.4)$$

The forward kinematics from the base to the tip is obtained by chaining all subsegment transformations. Denoting by N the total number of subsegments, the tip pose with respect to the base frame is obtained by

$${}^0\mathbf{T}_{\text{tip}} = \prod_{i=1}^N {}^{i-1}\mathbf{T}_i \quad (4.5)$$

4.4 Kineto-Static Modelling Framework

A purely kinematic model describes geometric reachability; however, practical feasibility requires static equilibrium under tendon forces, gravity, friction, and external loads. Under the quasi-static assumptions of Section 4.2.2, the equilibrium configuration of the two-segment TDCR is obtained by enforcing force–moment balance at each vertebra along the backbone. This section summarizes the vertebra-level load contributions, the friction model and tension propagation, the backbone elastic response, and the recursive equilibrium equations used in the kineto-static solver.

4.4.1 Loads and Contributions at Vertebra Level

The internal resultant force and moment transmitted from vertebra i toward the base are denoted by $\mathbf{F}_i$ and $\mathbf{M}_i$, respectively. Here, all local load contributions in the recursive equilibrium equations are expressed in a consistent vertebra-attached frame.

At each vertebra, the model accounts for tendon forces, the gravitational load associated with the vertebra and the corresponding backbone subsegment, externally applied tip wrenches propagated through the structure when present, and the elastic restoring moments generated by backbone bending and torsion.

For tendon c , the local tension at vertebra i is $T_{c,i}$. The tendon direction between vertebra $(i-1)$ and i is represented by the unit vector $\mathbf{u}_{i-1,i,c}$ in a consistent local frame. The tendon force contribution at vertebra i is

$$\mathbf{f}_{t,c,i} = T_{c,i}\mathbf{u}_{i-1,i,c} \quad (4.6)$$

and the corresponding moment about the vertebra center is

$$\mathbf{m}_{t,c,i} = \mathbf{p}_{t,c,i} \times \mathbf{f}_{t,c,i} \quad (4.7)$$

where $\mathbf{p}_{t,c,i}$ is the position vector from the vertebra center to the tendon guide of tendon c . The effect of gravity is modelled by lumping the vertebra mass m_d with the mass of the corresponding backbone subsegment at vertebra i .

The gravitational force expressed in the local frame of vertebra $(i-1)$ is

$$\mathbf{F}_{g,i} = \mathbf{R}_{i-1}^\top m_i \mathbf{g}, \quad m_i = m_d + m_b l_i, \quad \mathbf{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T \quad (4.8)$$

where $\mathbf{R}_{i-1}^\top$ is the rotation matrix used to express the gravitational load in frame $(i-1)$. External forces and moments applied at the tip are introduced at the top vertebra and propagated backward as internal resultants. Figure 4.4 illustrates the free-body diagram, showing tendon forces, gravity, and internal resultants at a representative vertebra of the discretized backbone.

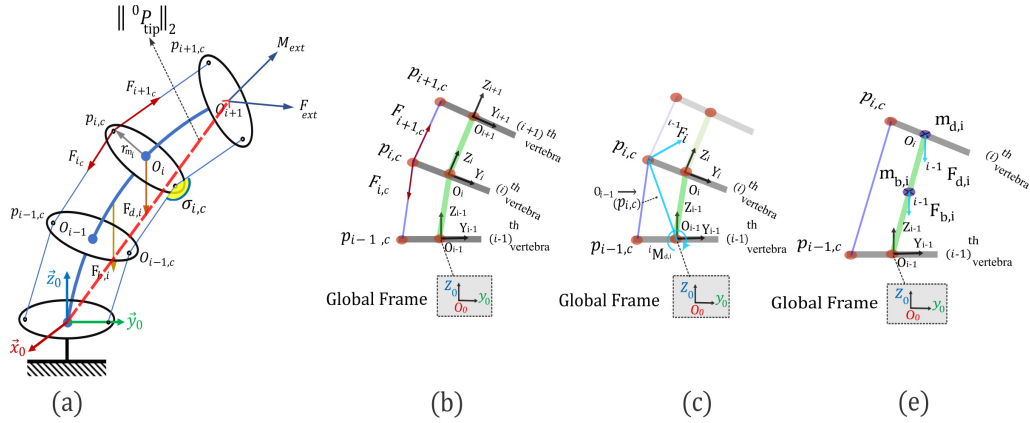


Figure 4.4 Free-body diagram of vertebra i and the adjacent backbone subsegment, showing tendon forces, gravity, and internal resultants.

4.4.2 Friction Modelling and Tension Propagation

Friction at tendon guides modifies tension transmission along the backbone. For numerical robustness in repeated equilibrium calculations, a vertebra-wise Coulomb model is adopted.

Let $\mathbf{u}_{i-1,i,c}$ and $\mathbf{u}_{i,i+1,c}$ be the tendon direction unit vectors at vertebra i on the incoming and outgoing segments, respectively. The local tendon deflection angle is

$$\sigma_{c,i} = \arccos(\mathbf{u}_{i-1,i,c} \cdot \mathbf{u}_{i,i+1,c}) \quad (4.9)$$

The corresponding normal force magnitude is defined as

$$n_{c,i} = T_{c,i} \sin\left(\frac{\sigma_{c,i}}{2}\right) \quad (4.10)$$

The friction magnitude at vertebra i is

$$f_{fr,c,i} = \mu n_{c,i} \quad (4.11)$$

where μ is the effective Coulomb friction coefficient. The tension transmitted to the next vertebra is updated as

$$T_{c,i+1} = T_{c,i} - f_{fr,c,i} \quad (4.12)$$

with the friction force acting in the direction opposite to the sliding motion of the tendon. This local formulation is consistent with the modelling assumptions of Section 4.2.2 and remains computationally tractable in the optimisation loops used later for feasible workspace evaluation.

4.4.3 Backbone Elasticity

Backbone elasticity provides the restoring moments that balance loads generated by tendons, gravity, and external forces. Denoting by $\kappa_{x,i}$ and $\kappa_{y,i}$ the curvature components of subsegment i in frame (i) , the bending moment vector is modelled as

$$\mathbf{M}_{b,i} = EI \begin{bmatrix} \kappa_{x,i} & \kappa_{y,i} & 0 \end{bmatrix}^T \quad (4.13)$$

where E is Young's modulus and I is the second moment of area of the backbone cross-section. When torsion is included, a linear torsional relation is added:

$$\mathbf{M}_{t,i} = GJ\tau_i\mathbf{e}_z \quad (4.14)$$

where G is the shear modulus, J the polar moment of area, and τ_i is the local twist rate, and $\mathbf{e}_z$ the local backbone axis.

4.4.4 Recursive Force and Moment Equilibrium

Static equilibrium is enforced in recursive Newton–Euler form, starting from the distal tip and propagating toward the base. Let $\mathbf{F}_i$ and $\mathbf{M}_i$ denote the internal resultant force and moment at vertebra i , expressed in the chosen local frame. The corresponding force and moment recursions are written as

$$\mathbf{F}_{i-1} = \mathbf{F}_i + \mathbf{F}_{g,i} + \sum_c \mathbf{f}_{t,c,i} + \mathbf{F}_{ext,i} \quad (4.15)$$

$$\mathbf{M}_{i-1} = \mathbf{M}_i + \mathbf{M}_{g,i} + \sum_c \mathbf{m}_{t,c,i} + \mathbf{M}_{\text{ext},i} + \mathbf{M}_{b,i} + \mathbf{M}_{t,i} \quad (4.16)$$

where $\mathbf{F}_{\text{ext},i}$ is nonzero only when an external force is applied at, or propagated to, vertebra i . Here, $\mathbf{M}_{g,i}$ denotes the moment of the gravitational force about the vertebra center, $\mathbf{m}_{t,c,i}$ is the moment generated by tendon c at vertebra i , $\mathbf{M}_{\text{ext},i}$ is the external moment propagated to vertebra i , and $\mathbf{M}_{b,i}$ and $\mathbf{M}_{t,i}$ are the elastic bending and torsional moments defined in Eqs. (4.13) and (4.14), respectively.

For a given set of tendon tensions within the prescribed bounds, the recursive equilibrium equations are solved along the discretised backbone to obtain a static configuration that is both kinematically consistent with the PCC model of Section 4.3 and in force–moment equilibrium. The resulting kineto-static solver is used in Section 4.5 for feasible static workspace analysis.

4.5 Study A: Feasible Static Workspace and Tendon-Force Optimization

This section presents Study A, in which tendon force distributions are optimized for a fixed mechanical design to characterize and maximize the feasible static workspace (FSW) of the two-segment TDCR under load. The kineto-static model of Section 4.4 is used as the feasibility constraint.

4.5.1 Definition

The feasible static workspace is defined here as the set of tip positions that can be reached and maintained in static equilibrium while satisfying tendon-force bounds and the kineto-static equilibrium equations. Let $\mathbf{P}_{\text{tip}}(\mathbf{T}) \in \mathbb{R}^3$ denote the tip position obtained by solving the kineto-static model for a tendon tension vector $\mathbf{T} \in \mathbb{R}^8$.

$$0 \leq T_c \leq 1.6\text{N}, \quad c = 1, \dots, 8 \quad (4.17)$$

The kineto-static solver enforces force–moment equilibrium at each vertebra together with the elastic relations and geometric consistency. Collecting these conditions into a residual vector $\Theta(\mathbf{T}, \mathbf{S})$, where $\mathbf{S}$ contains the curvature and twist parameters of all subsegments, the feasible static workspace is written as

$$\mathcal{W}_{\text{FSW}} = \{ \mathbf{P}_{\text{tip}}(\mathbf{T}) \in \mathbb{R}^3 \mid \exists \mathbf{S}, \Theta(\mathbf{T}, \mathbf{S}) = \mathbf{0}, T_{\min} \leq T_c \leq T_{\max} \} \quad (4.18)$$

In practice, for a given $\mathbf{T}$ the solver is run once; if it converges to a physically admissible equilibrium (no loss of tendon tension, no geometric inconsistency), the corresponding $\mathbf{P}_{\text{tip}}(\mathbf{T})$ is accepted as a feasible point. Otherwise, $\mathbf{T}$ is discarded and does not contribute to $\mathcal{W}_{\text{FSW}}$.

4.5.2 Scalar Metric and Optimization Problem

To represent workspace extent by a single scalar quantity consistent with the subsequent design studies, Study A uses the static reach under the prescribed external load as its optimisation metric. For a feasible tendon-force vector $\mathbf{T}$, the static reach is defined as the Euclidean norm of the tip position, $r(\mathbf{T}) = \|\mathbf{p}_{\text{tip}}(\mathbf{T})\|_2$.

The tendon-force optimisation problem for maximising static reach can be written as

$$\min_{\mathbf{T}} J(\mathbf{T}) = \frac{1}{r(\mathbf{T})}, \quad \text{s.t. } \Theta(\mathbf{T}, \mathbf{S}) = \mathbf{0}, \quad T_{\min} \leq T_c \leq T_{\max} \quad (4.19)$$

If the kineto-static solver fails to converge or returns an infeasible configuration for a candidate $\mathbf{T}$, a large penalty value is assigned to $J(\mathbf{T})$. In this way, infeasible candidates are automatically rejected by the optimizer.

The mapping $\mathbf{T} \mapsto \mathbf{p}_{\text{tip}}(\mathbf{T})$ is highly nonlinear, due to gravity, friction, and backbone elasticity, so $J(\mathbf{T})$ is nonconvex. A population-based genetic algorithm is therefore adopted: each individual encodes a candidate tendon-force vector, the kineto-static solver evaluates the corresponding equilibrium configuration, and the objective $J(\mathbf{T})$ guides selection and variation in the population.

4.5.3 Numerical Results

For the baseline robot configuration of Table 4.1, Study A evaluates both the feasible static workspace and the effect of tendon-force optimisation under the prescribed loading condition.

In a first experiment, 100 tendon-force vectors $\mathbf{T} \in \mathbb{R}^8$ are randomly generated within the actuator bounds. The kineto-static solver is applied to each candidate, and the feasible tip positions are collected according to the criteria defined in Section 4.5.2. The resulting set of admissible tip positions is shown in Figure 4.5.

In a second experiment, the genetic algorithm described above is executed to identify the tendon-force vector that maximizes static reach. The evolution of the objective function $J(\mathbf{T})$ over the optimization iterations is reported in Figure 4.6.

The blue curve reports the best value of the objective function $J(\mathbf{T})$ in each generation, while the orange curve shows the corresponding population mean. Over the first few gener-

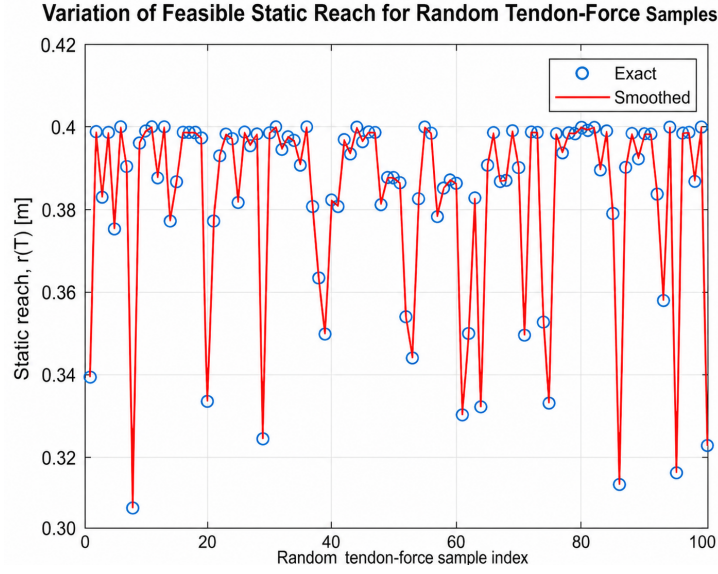


Figure 4.5 Feasible static workspace obtained from 100 randomly sampled tendon-force vectors.

ations the best and mean values of $J(\mathbf{T})$ both decrease and then stabilise around $J \approx 2.50$, indicating that the genetic algorithm rapidly converges towards a set of tendon-force patterns that achieve near-maximal static reach for the chosen robot configuration and external load.

Figure 4.7 shows the equilibrium configuration corresponding to the optimized tendon-force vector under a prescribed downward tip force of 0.1 N together with an external tip moment $\mathbf{M}_{\text{ext}} = [0.01 \ 0.01 \ 0]^T \text{ N.m}$.

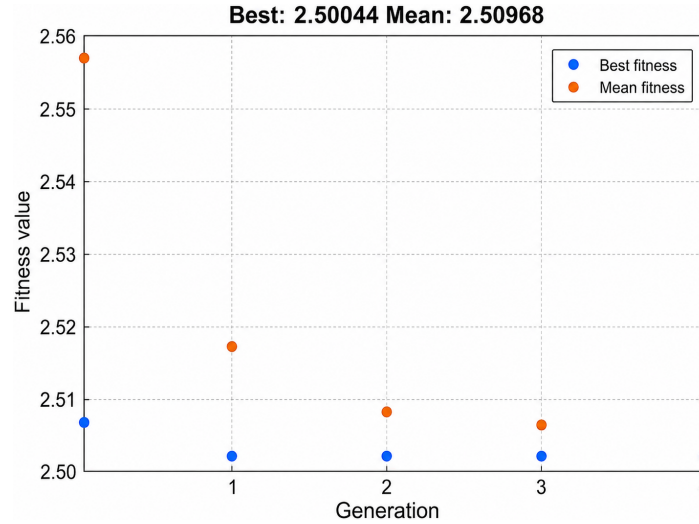


Figure 4.6 The trend of objective function changes during tendon-force optimization.

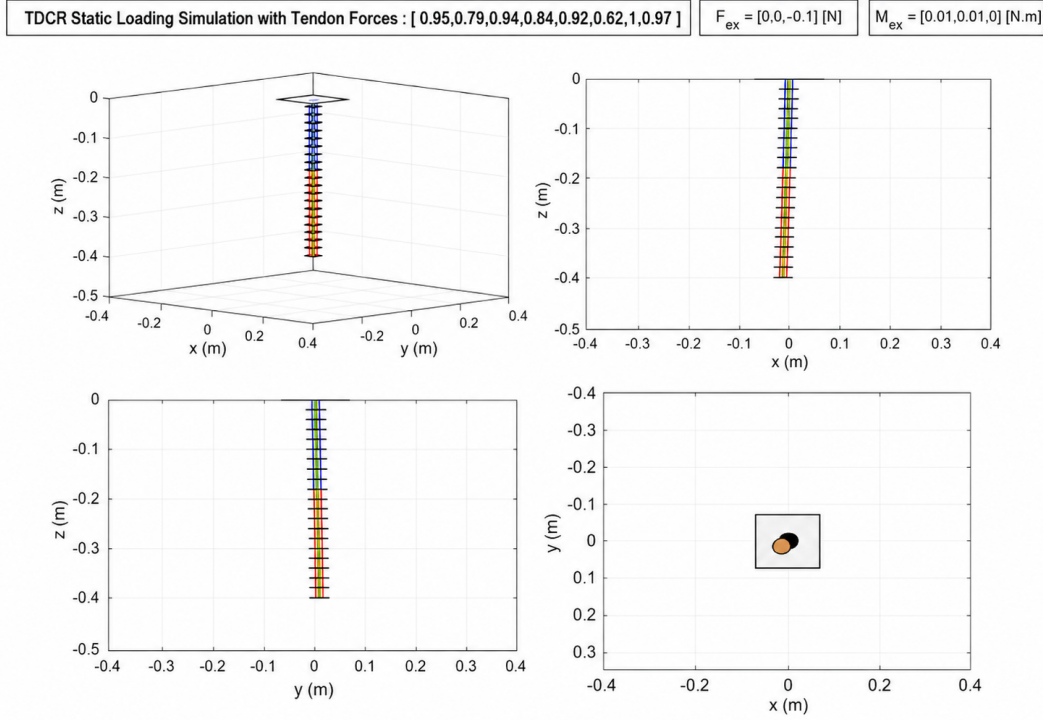


Figure 4.7 Optimal equilibrium configuration under vertical tip load.

4.6 Study B: Multi-Objective Dimensional Synthesis of Tendon Routing

Section 4.5 evaluated the feasible static workspace (FSW) by optimizing tendon-force distributions for a fixed robot geometry. That study demonstrated how actuation redundancy can enhance static reachability under load, but it did not investigate the influence of tendon routing geometry itself. In practice, tendon placement on the vertebrae directly affects bending leverage, frictional transmission, and the mechanical effort required to maintain equilibrium. This section therefore extends the optimization problem from the actuator space to the geometric design space.

The goal is to identify tendon routing layouts that achieve a favorable trade-off between feasible static workspace and mechanical effort, while satisfying manufacturability requirements and tendon non-interference constraints. The same PCC-based kineto-static solver introduced in Section 4.4 is employed so that all candidate designs are evaluated within a consistent equilibrium framework.

4.6.1 Design Variables

The dimensional synthesis is performed on the prototype-matched two-segment TDCR, which has seven vertebrae and four tendons per segment. In Study B, the structural properties of the backbone and vertebrae are fixed, and the optimisation is carried out only over the tendon-routing geometry. The design vector is defined as

$$\xi = \begin{bmatrix} r_1 & r_2 & \alpha \end{bmatrix}^T \quad (4.20)$$

where r_1 and r_2 denote the tendon routing radii in Segments 1 and 2, respectively, and α represents the inter-segment angular offset between the tendon patterns (Figure 4.8(a)).

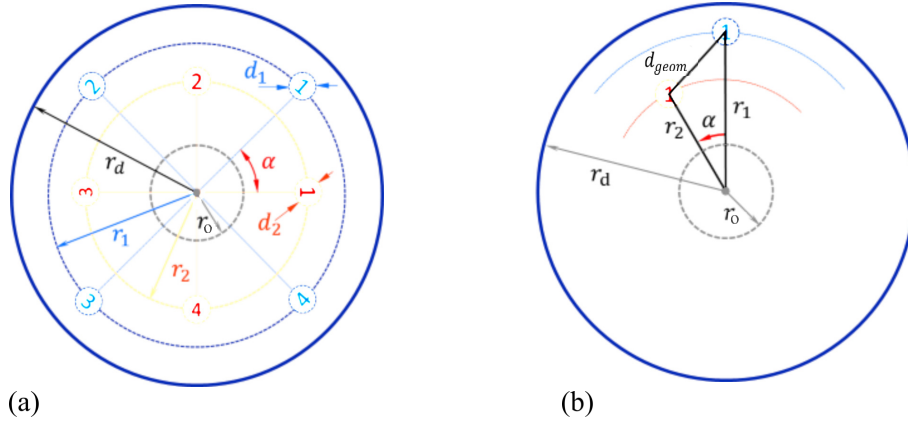


Figure 4.8 (a) Four-tendon arrangement on the vertebra for both segments. (b) Geometric constraint ensuring minimum tendon spacing.

4.6.2 Objective Functions and Performance Indicators

For a given design ξ and tendon-force vector $\mathbf{T}$, the kineto-static solver computes the static equilibrium configuration and the corresponding internal force and moment resultants along the discretized backbone. For each vertebra i , let

- $\mathbf{P}_i \in \mathbb{R}^3$ denote its position in the base frame.
- $\mathbf{q}_i = [\kappa_i, \varphi_i, \rho_i]^T$ collects the curvature magnitude, bending-plane angle, and twist rate of subsegment i .
- $\mathbf{F}_{\text{net},i}$ and $\mathbf{M}_{\text{net},i}$ be the net internal force and moment at vertebra i returned by the kineto-static solver.

A quasi-static work metric is then defined from the net force and moment contributions along the robot:

$$W_{\text{disp}}(\xi, \mathbf{T}) = \sum_{i=1}^N \mathbf{F}_{\text{net},i} \cdot \mathbf{p}_i^T \quad (4.21)$$

$$W_{\text{rot}}(\xi, \mathbf{T}) = \sum_{i=1}^N \mathbf{M}_{\text{net},i} \cdot \mathbf{q}_i^T \quad (4.22)$$

$$W_{\text{tot}}(\xi, \mathbf{T}) = |W_{\text{disp}}(\xi, \mathbf{T})| + |W_{\text{rot}}(\xi, \mathbf{T})| \quad (4.23)$$

A lower value of W_{tot} indicates a more efficient design for the same loading condition and tendon-force realization. The second quantity is the static reach, defined as the Euclidean norm of the tip position obtained from the equilibrium configuration:

$$\rho(\xi, \mathbf{T}) = \|\mathbf{p}_{\text{tip}}(\xi, \mathbf{T})\|_2 \quad (4.24)$$

For statistical analysis and comparison across runs, Pareto-optimal designs are evaluated using the performance indicators

$$C_1(\xi, \mathbf{T}) = \frac{1}{W_{\text{tot}}(\xi, \mathbf{T})}, \quad C_2(\xi, \mathbf{T}) = \rho(\xi, \mathbf{T}) \quad (4.25)$$

4.6.3 Geometric Bounds and Non-Interference Constraint

The design search is restricted by geometric feasibility and manufacturability. The routing radii are bounded within the admissible radial range of the vertebra geometry:

$$r_o \leq r_1, r_2 \leq r_d \quad (4.26)$$

This constraint reflects a fundamental design trade-off, as larger routing radii increase the effective moment arm and actuation efficiency, while also being limited by vertebra size and structural integrity.

The angular offset is constrained to the practical range adopted in the prototype study:

$$\alpha_{\min} \leq \alpha \leq \alpha_{\max}, \quad 0^\circ \leq \alpha \leq 45^\circ \quad (4.27)$$

To avoid overlap between tendon guide channels of the two routing patterns, a nonlinear non-interference constraint is enforced. The center-to-center distance between corresponding channels is

$$d_{\text{geom}}(r_1, r_2, \alpha) = \sqrt{r_1^2 + r_2^2 - 2r_1r_2 \cos \alpha} \quad (4.28)$$

and must satisfy

$$d_{\text{geom}}(r_1, r_2, \alpha) \geq d_{\text{min}}, \quad d_{\text{min}} = \frac{d_1 + d_2}{2} + d_t \quad (4.29)$$

where d_1 and d_2 are the tendon hole diameters and d_t denotes the minimum edge-to-edge safety clearance (Figure 4.8(b)). This constraint is implemented as a nonlinear inequality within the optimization and eliminates geometrically infeasible tendon layouts.

4.6.4 Optimization Procedure

The bi-objective optimization then seeks to reduce the total work metric while increasing the feasible static reach. Using the minimization convention of the solver, the objective vector is written as:

$$\begin{aligned} & \min_{\xi} [W_{\text{tot}}(\xi, \mathbf{T}), -\rho(\xi, \mathbf{T})] \\ & \text{subject to :} \\ & r_o \leq r_m \leq r_d, \quad m \in \{1, 2\}, \\ & \alpha_{\text{min}} \leq \alpha \leq \alpha_{\text{max}}, \\ & d_{\text{geom}}(r_1, r_2, \alpha) \geq d_{\text{min}}. \end{aligned} \quad (4.30)$$

4.6.5 Results and Design Guidelines

The problem is solved for 100 randomized tendon-force samples in order to account for variability in actuation patterns. For each realization, the resulting non-dominated solutions define a Pareto front in the objective space. For each candidate design in the population, the tendon routing geometry is updated, the kineto-static solver is executed once, and the indicators (C_1, C_2) are computed from the resulting equilibrium configuration.

Representative Pareto fronts obtained from two distinct tendon-force realisations are shown in Figures 4.9(a) and 4.9(b), illustrating the typical trade-off between mechanical work and static reach.

To extract design trends, Pareto-optimal solutions are grouped according to the angular offset α using 5° bins in the interval $[0^\circ, 45^\circ]$. For each bin, the number of samples and the mean and variance of r_1 , r_2 , and α are computed. The same statistical quantities are evaluated for the performance indicators C_1 (inverse work) and C_2 (static reach). The results are summarized in Table 4.2 and Figures 4.10–4.18.

A higher concentration of solutions is observed for $\alpha < 30^\circ$. However, small angular offsets increase the risk of tendon interference during large bending and reduce compatibility

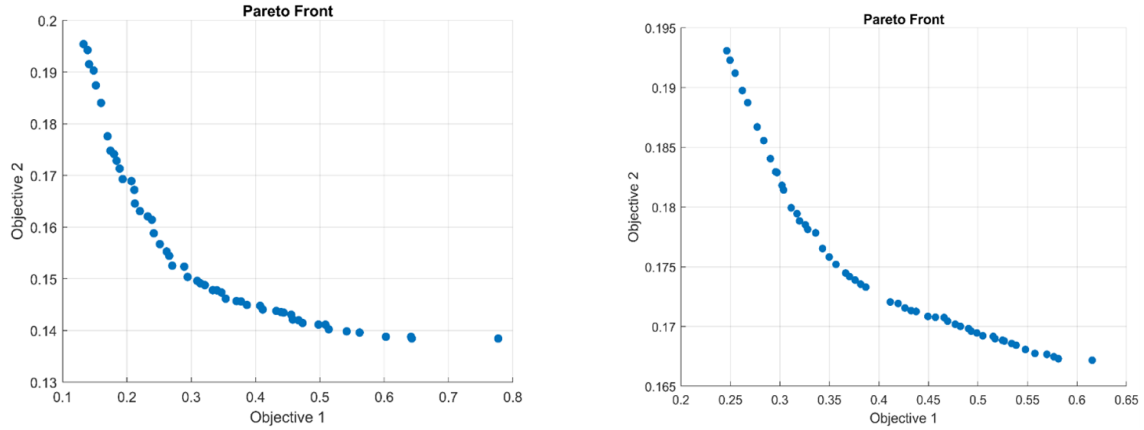


Figure 4.9 (a) Pareto front diagram, 14th iteration. (b) Pareto front diagram, 61st iteration.

Table 4.2 Statistical summary of Pareto-optimal designs grouped by α -range across all optimisation runs.

α range (deg)	Samples	Mean r_1 (mm)	Mean r_2 (mm)	Mean α (deg)	Mean C_1 (J ⁻¹)	Mean C_2 (mm)	Var. r_1 (mm)	Var. r_2 (mm)	Var. α (deg)	Var. C_1 (J ⁻¹)	Var. C_2 (mm)
0–5	454	19.64	23.34	2.79	1.58	255.36	2.19	0.39	1.52	488.59	3.97
5–10	672	20.42	23.30	7.59	1.65	257.06	2.13	0.46	1.45	662.97	3.75
10–15	693	20.23	23.33	12.48	1.48	258.40	2.29	0.20	1.45	396.64	2.95
15–20	612	20.28	23.31	17.37	1.45	260.32	2.24	0.21	1.46	415.43	2.31
20–25	374	20.33	23.30	22.30	1.41	261.97	2.22	0.22	1.36	401.41	1.60
25–30	157	20.05	23.35	27.13	1.44	262.95	2.39	0.18	1.39	535.29	1.36
30–35	50	21.58	23.34	32.30	2.07	263.69	1.86	0.14	1.46	1255.66	0.85
35–40	37	22.81	23.23	37.22	3.35	264.06	0.63	0.23	1.60	1464.18	0.76
40–45	40	21.79	23.05	43.46	2.50	264.54	2.08	0.51	1.47	1467.40	0.30

with the prototype mounting. For this reason, the design selection does not rely solely on solution density.

Across the different α ranges, high-performing solutions tend to place the tendon channels close to the maximum admissible radii. The second segment typically adopts a slightly larger routing radius than the first one.

For $\alpha \geq 30^\circ$ range, the mean inverse work increases (Figure 4.15) while workspace variability decreases, indicating stable reach performance across tendon-force realizations (see Figure 4.18). The increase in variance of inverse work is offset by the improved mean value and reduced workspace dispersion, resulting in a more favorable trade-off (see Figure 4.16).

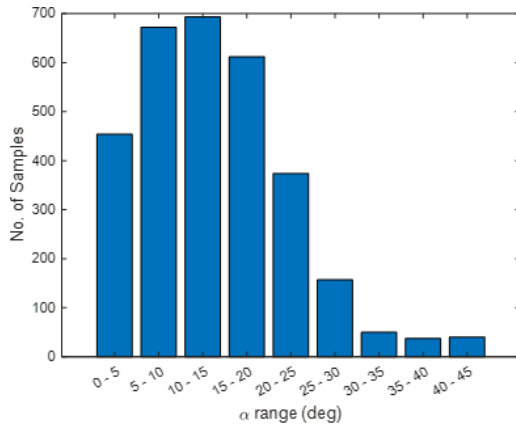


Figure 4.10 Frequency histogram of the data as a function of α angle divisions.

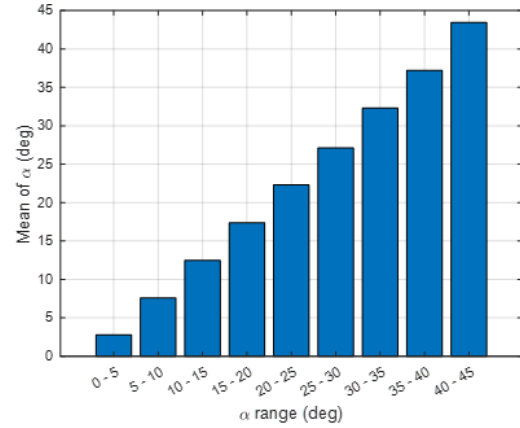


Figure 4.11 Average α angle for the optimal samples across the angular divisions.

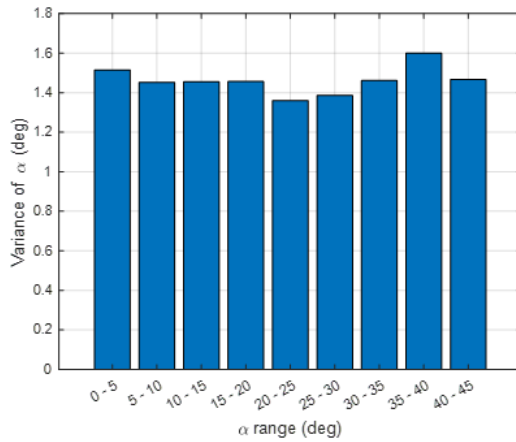


Figure 4.12 Variance in the α angle within each division.

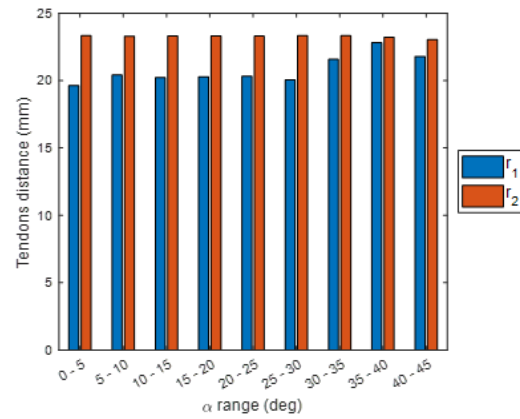


Figure 4.13 Average radial distance of the tendons from the center of the vertebra for each segment.

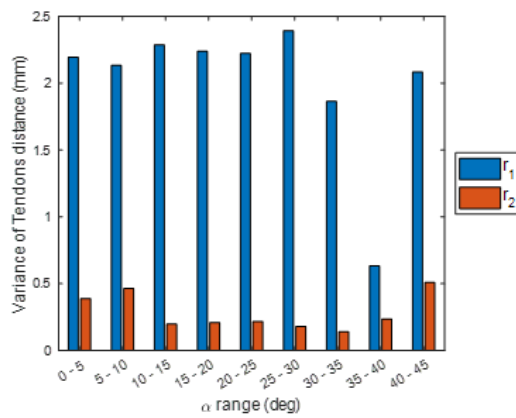


Figure 4.14 Variance in radial tendon distances for each segment.

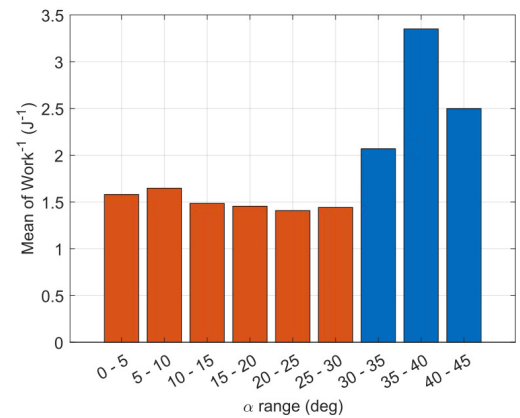


Figure 4.15 Inverse work across the range of α angle variations.

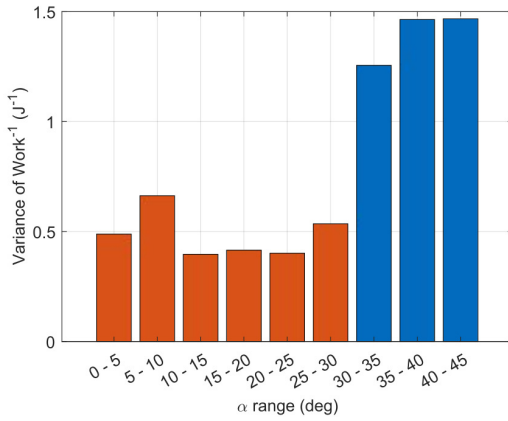


Figure 4.16 Variance of inverse work across the range of α angle variations.

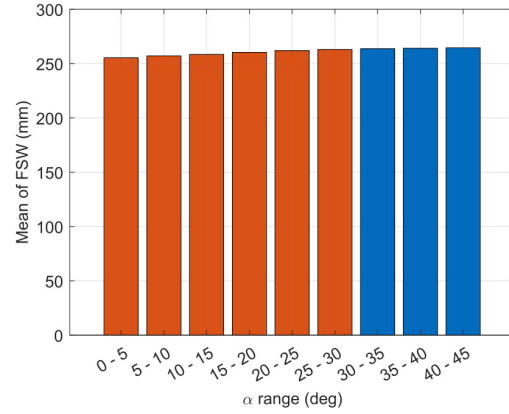


Figure 4.17 Static feasible workspace across the range of α angle variations.

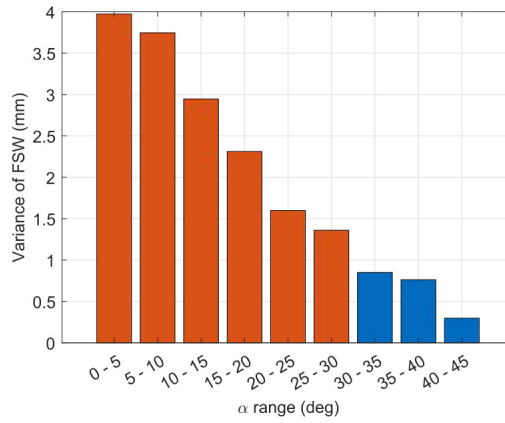


Figure 4.18 Variance of static feasible workspace across the range of α angle variations.

Based on the Pareto distribution, statistical evaluation, geometric constraint, and mechanical considerations, the routing parameters selected for the subsequent studies are $r_1 = 21$ mm, $r_2 = 22.5$ mm, $\alpha = 45^\circ$, which provide a balance between workspace expansion and reduced mechanical work.

4.7 Chapter Summary

This chapter presented the modelling framework used to analyse static equilibrium in the two-segment TDCR considered in this thesis. The mechanical structure and tendon-routing arrangement were first introduced, and a discretised piecewise constant curvature model was formulated to relate the configuration variables to the end-effector pose. On this basis, a kineto-static formulation was developed to describe equilibrium under tendon tensions, gravity, vertebra-level friction, backbone elasticity, and external loads.

Within this framework, the feasible static workspace was defined and analysed for a fixed robot design through tendon-force optimisation (Study A). The same formulation was then extended to a design optimisation problem in which the tendon-routing radii and the inter-segment angular offset were treated as design variables (Study B). The Pareto results, together with the statistical analysis of the obtained solutions, provided practical design indications for the robot studied here, including a preference for larger inter-segment offsets ($\alpha \geq 30^\circ$) and tendon routing close to the outer admissible radii.

The results presented in this chapter remain specific to the robot architecture considered here, namely a two-segment TDCR with fixed segment lengths and a fixed number of vertebrae. The chapter also showed a practical limitation of the kineto-static approach. Although the model is suitable for design analysis and static feasibility evaluation, it is not convenient for direct use in real-time inverse kinematics or online trajectory generation, since each evaluation requires repeated numerical solutions of the equilibrium problem. For this reason, Chapter 5 shifts the computationally demanding kineto-static analysis to an offline stage, where feasible configurations and performance indicators are precomputed, while online execution is based on fast selection or learning-based estimation followed by local refinement.

Chapter 5

From Hybrid Task-Space Planning to Offline Capability-Map for Tendon-Driven Continuum Robots

5.1 Motivation and Scope of the Chapter

Chapter 4 established the model-based description of the robot. A kineto-static formulation was developed to evaluate equilibrium under tendon-force limits, backbone elasticity, friction between tendons and vertebrae, and external loading. This allowed the feasible static workspace to be characterized explicitly rather than assumed. The routing geometry was then optimized through a bi-objective study, leading to the selection of a configuration defined by $r_1 = 21$ mm, $r_2 = 22.5$ mm, $\alpha = 45^\circ$. This geometry serves as the fixed reference design for all developments that follow.

Although the model provides a consistent physical description, it is computationally expensive. Evaluating a single configuration requires solving the equilibrium problem across the 14 vertebra-level elements of the discretised robot. For design analysis, this cost is acceptable. For trajectory execution, however, repeatedly solving the constrained equilibrium problem at every waypoint quickly becomes impractical, especially when the path is finely discretized.

Trajectory generation remains challenging because TDCRs are hyper-redundant and highly nonlinear systems. In practice, inverse kinematics is solved through constrained optimization to enforce actuator limits and feasibility. However, convergence may become unreliable near workspace boundaries, and computational cost increases rapidly with tra-

jectory length. As a result, purely online optimization is rarely compatible with real-time execution.

For this reason, the present chapter does not modify the physical model introduced earlier, but reconsiders how it is used during execution. Two complementary strategies are developed. The first combines a supervised neural network, used to provide fast inverse-kinematics initialization, with a constrained refinement step that enforces the adopted bound constraints and improves waypoint accuracy. The second adopts an offline capability-map approach, where feasibility is evaluated in advance and organized for efficient runtime selection. In both cases, the objective is to retain the physical consistency established in Chapter 4 while reducing the computational cost of motion execution.

5.2 Task-Space Trajectory and Discretisation

This section formulates the task-space trajectory generation problem in the vertebra-wise PCC setting adopted throughout the TDCR part of the thesis, using the online inverse-kinematics pipeline implemented for trajectory execution.

A desired end-effector motion is specified in task space by the pair $(\mathbf{p}_d(t), \mathbf{R}_d(t))$, where $\mathbf{p}_d(t) \in \mathbb{R}^3$ denotes the desired position and $\mathbf{R}_d(t) \in SO(3)$ denotes the desired orientation:

$$x_d(t) = [\mathbf{p}_d(t), \mathbf{R}_d(t)], \quad t \in [0, T]. \quad (5.1)$$

In the present implementation, t is used as a trajectory parameter for sampling the reference path rather than as a physical time variable. The position reference is selected from one of four path families, circular, elliptical, helical, or butterfly, while the orientation reference is prescribed through one of four orientation-lock modes: fixed tilt around the Y-axis, lock about the X-axis, lock about the Y-axis, or lock about the Z-axis.

For execution, the continuous reference is discretised into N waypoints,

$$\left\{ x_d^k \right\}_{k=1}^N, \quad x_d^k = [\mathbf{p}_d^k, \mathbf{R}_d^k]. \quad (5.2)$$

Each waypoint is then encoded as a desired homogeneous transformation,

$$\mathbf{T}_{\text{des}}^k = \begin{bmatrix} \mathbf{R}_d^k & \mathbf{p}_d^k \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (5.3)$$

The robot configuration follows the same discretised piecewise constant-curvature representation introduced in Chapter 4. For the two-segment robot considered here, the backbone is represented by 14 subsegments, and the optimisation variables of subsegment i are defined

as

$$\mathbf{q}_i = \begin{bmatrix} \kappa_i & \varphi_i & \tau_i \end{bmatrix}^T, \quad i = 1, \dots, N_s, \quad (5.4)$$

where κ_i is curvature magnitude, φ_i is the bending-plane angle, and τ_i is the twist rate of the subsegment. For the two-segment robot considered here, $N_s = 14$, so that

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T & \dots & \mathbf{q}_{14}^T \end{bmatrix}^T \in \mathbb{R}^{42}. \quad (5.5)$$

Given $\mathbf{q}$, the forward kinematics $\mathbf{T}(\mathbf{q})$ are obtained by chaining the PCC subsegment transformations, yielding the predicted end-effector position $\mathbf{p}(\mathbf{q})$ and orientation $\mathbf{R}(\mathbf{q})$.

At each waypoint k , inverse kinematics is solved through the following bound-constrained local optimisation problem:

$$\begin{aligned} \min_{\mathbf{q}_k} \quad & \left\| \mathbf{p}(\mathbf{q}_k) - \mathbf{p}_d^k \right\|_2 + \omega \left\| \mathbf{R}(\mathbf{q}_k) - \mathbf{R}_d^k \right\|_F^2 \\ \text{s.t.} \quad & 0 \leq \kappa_i \leq 25 \text{ m}^{-1}, \\ & 0 \leq \varphi_i \leq 2\pi \text{ rad}, \\ & -6 \leq \tau_i \leq 6 \text{ rad/m}. \end{aligned} \quad (5.6)$$

The orientation weight factor $\omega = 0.05$ was selected empirically through extensive testing across 16 trajectory–orientation combinations.

Because the TDCR is kinematically redundant, solving each waypoint independently can lead to discontinuous jumps in configuration space. Continuity is therefore promoted through two complementary mechanisms. First, each waypoint is warm-started from the previously accepted solution $\mathbf{q}_{k-1}$. Second, a six-start strategy is used, consisting of the previous accepted solution, a nominal default guess, and four random initialisations sampled within the bound constraints. Among the converged candidates, the solution with the lowest objective value is retained.

A waypoint is accepted only if the optimised solution satisfies

$$\varepsilon_p = \left\| \mathbf{p}(\mathbf{q}_k) - \mathbf{p}_d^k \right\| < 10^{-2} \text{ m}. \quad (5.7)$$

Otherwise, the waypoint is discarded. In the present implementation, this acceptance check is based on positional accuracy only.

5.3 Tendon Length Variation Profiles

While the previous sections analyse trajectory tracking in task space, the TDCR is physically actuated through tendon-length variations. Therefore, the configuration sequence $\{\mathbf{q}_k\}$ obtained from the hybrid pipeline is also expressed in terms of tendon commands.

The tendon-length variation is derived from the discretised PCC deformation between two consecutive vertebrae $(i-1, i)$, illustrated in Figure 5.1(a). As the backbone bends, the tendon displacement depends on the relative orientation between the bending direction and the tendon position in the cross-section, as shown in Figure 5.1(b–c).

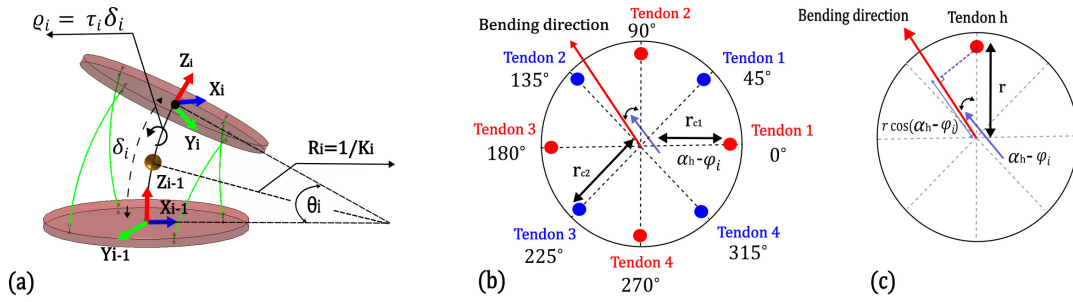


Figure 5.1 Geometric interpretation of tendon-length variation. (a) Deformation between two consecutive vertebrae. (b) Tendon placement in the cross-section. (c) Projection of the tendon position onto the bending direction.

For each waypoint k , the configuration vector $\mathbf{q}_k \in \mathbb{R}^{42}$ provides the vertebra-level parameters $\{\kappa_i^k, \varphi_i^k, \tau_i^k\}$. The incremental tendon-length variation contributed by subsegment i is

$$\Delta l_h(i, k) = r \delta_i \left(\kappa_i^k \cos(\alpha_h - \varphi_i^k) + \tau_i^k \right), \quad r = r_{c1} \text{ or } r_{c2}, \quad h = 1, 2, 3, 4. \quad (5.8)$$

Let δ_i denote the uniform subsegment length, segment length divided by the number of vertebrae in that segment, $\delta_i = 132.63/7 = 0.01895$ m, the tendon routing radii $r_{c1} = 21$ mm and $r_{c2} = 22.5$ mm, and α_h the angular position of tendon h in the corresponding cross-section. The total tendon displacement at waypoint k is obtained by summing the contributions of the subsegments belonging to each segment

$$\Delta l_{m,h}^k = \sum_{i \in \zeta_m} \Delta l_h(i, k), \quad m \in \{1, 2\}, \quad h = 1, 2, 3, 4. \quad (5.9)$$

The index sets $\zeta_1 = \{1, \dots, 7\}$ and $\zeta_2 = \{8, \dots, 14\}$ denote the vertebrae belonging to Segment 1 and Segment 2, respectively.

5.4 Offline IK Data Generation for Supervised Learning

While Section 5.2 describes the waypoint-wise inverse-kinematics procedure used for online trajectory execution, the same formulation is now employed offline for dataset generation. The objective is to construct a collection of accepted pose–configuration pairs that remain consistent with the bound-constrained optimisation problem of Eq. (5.6) and the acceptance criterion of Eq. (5.7).

For each selected trajectory family and orientation-lock mode, the continuous task-space reference defined in Eq. (5.1) is discretised into N waypoints as in Eq. (5.2). In the experiments reported later in this chapter, $N = 1000$ waypoints are generated per trajectory. The considered trajectory set includes circular, elliptical, helical, and butterfly paths, each combined with the orientation prescriptions introduced in Section 5.2.

To store each accepted waypoint in a compact and implementation-consistent form, the input pose is encoded as a 12-dimensional vector composed of the desired position and the flattened rotation matrix,

$$\mathbf{x}^k = \left[(\mathbf{p}_d^k)^T \quad \text{vec}(\mathbf{R}_d^k)^T \right]^T \in \mathbb{R}^{12}, \quad (5.10)$$

where $\text{vec}(\cdot)$ denotes row-wise flattening of the 3×3 matrix. The corresponding supervised target is the configuration vector $\mathbf{q}_k \in \mathbb{R}^{42}$ obtained by solving Eq. (5.6) and stacking the subsegment variables as in Eq. (5.5). The resulting dataset is denoted by

$$\mathcal{D} = \left\{ (\mathbf{x}^k, \mathbf{q}_k) \right\}_{k=1}^{N_{\text{acc}}}, \quad (5.11)$$

where N_{acc} denotes the total number of accepted waypoints retained after applying the criterion in Eq. (5.7). A practical aspect of this procedure is redundancy. Because multiple local solutions may exist for the same waypoint, and because branch switching along a trajectory can lead to discontinuous configuration sequences, the dataset is generated using the same continuity-promoting strategy already described in Section 5.2. Each waypoint is initialised using the previously accepted solution whenever available, and a small multi-start set is evaluated by combining this warm-start, a nominal default guess, and four random initialisations sampled within the bound constraints. The best converged solution is retained, then screened using Eq. (5.7). This produces a dataset of accepted samples that favours locally consistent solutions along each reference trajectory, which is important for subsequent learning and trajectory execution.

The implemented offline data-generation procedure is summarised in Algorithm 5.1.

Algorithm 5.1 Offline generation of pose–configuration pairs for supervised learning

Input: trajectory family and orientation-lock mode; waypoint count N ; bounds in Eq. (5.6); tolerance in Eq. (5.7).

Output: dataset $\mathcal{D}$ containing pose–configuration pairs.

1. Generate the discretised reference waypoints $\{\mathbf{p}_d^k, \mathbf{R}_d^k\}_{k=1}^N$ and build $\mathbf{T}_{\text{des}}^k$ as in Eq. (5.3).
 2. Initialise a nominal default guess $\mathbf{q}^{(0)}$ and set the warm-start $\mathbf{q}_{ws} \leftarrow \mathbf{q}^{(0)}$.
 3. For $k = 1, \dots, N$:
 - (a) Construct multi-start set $\mathcal{G}$ as in Section 5.2, previous solution, default, and four random samples within bounds.
 - (b) Solve the constrained IK problem Eq. (5.6) from each $\mathbf{q}_{\text{init}} \in \mathcal{G}$ and retain the best converged solution $\mathbf{q}_k$.
 - (c) Compute the end-effector position $\mathbf{p}(\mathbf{q}_k)$ via forward kinematics and evaluate the waypoint error $\varepsilon_p = \|\mathbf{p}(\mathbf{q}_k) - \mathbf{p}_d^k\|$.
 - (d) If ε_p satisfies Eq. (5.7), store $(\mathbf{x}^k, \mathbf{q}_k)$ in $\mathcal{D}$ and update $\mathbf{q}_{ws} \leftarrow \mathbf{q}_k$. Otherwise, discard waypoint k .
 4. Repeat Steps 1–3 for all selected trajectory families and orientation-lock modes.
-

5.5 Supervised Learning Model and Training Setup

The supervised learning component provides a fast pose-to-configuration initialisation by approximating the inverse mapping from the 12-dimensional task-space input Eq. (5.10) to the 42-dimensional configuration vector defined in Section 5.2. The model is trained on the offline-generated dataset of Section 5.4, ensuring consistency with the discretised PCC representation and the bound constraints of the implemented pipeline. Two feed-forward architectures are evaluated, MLP and DNN, as summarised in Table 5.1. Before training, inputs and outputs are normalised to $[0, 1]$ using min–max scaling computed over the loaded dataset. Samples are randomly shuffled, and the training routine follows the implemented script: a training subset is used to update network parameters while a separate validation subset is used to monitor convergence during training. Training is performed with the Adam optimiser, learning rate 10^{-3} , batch size 32, for 200 epochs, and the trained network is saved together with the normalisation parameters to ensure consistent inference.

After training, the network is used only as an initial guess provider. For each waypoint, the predicted $\hat{\mathbf{q}}_k$ is passed to the constrained refinement stage of Section 5.6, which remains responsible for meeting the waypoint tracking objective under the bound constraints. This separation of roles allows learning to improve initialisation quality and runtime behaviour, while constraint satisfaction and final tracking accuracy remain enforced by the optimisation layer. In the following, the trained regressor is denoted by $f_\theta(\cdot)$, where θ collects the learned network parameters.

Table 5.1 Summary of supervised learning architectures and training setup.

Parameter	MLP	DNN
Hidden layers	2	3
Neurons per layer	64–64	128–128–64
Activation	ReLU	ReLU
Output layer	Linear (42 neurons)	Linear (42 neurons)
Input dimension	12 (position + $\text{vec}(\mathbf{R})$)	12
Output dimension	42 (configuration vector $\mathbf{q}$)	42
Normalisation	Min–max scaling to $[0, 1]$	Min–max scaling to $[0, 1]$
Dataset split	Learning: 67.5%, Validation: 12.5%, Test: 20%	Same
Optimizer	Adam (learning rate 10^{-3})	Same
Batch size	32	32
Epochs	200	200
Loss function	Mean-squared error (MSE)	Same
Role in pipeline	Fast initial guess provider for refinement (Section 5.6)	Same

5.6 Hybrid Online Inverse-Kinematics Pipeline

The hybrid pipeline combines the supervised model of Section 5.5 with the waypoint-wise constrained optimisation of Section 5.2 in order to generate accepted trajectories through neural initialisation followed by local refinement. For each waypoint x_d^k , the network provides a fast initial guess $\hat{\mathbf{q}}_k$, which is then refined by the constrained solver to satisfy the bound constraints and meet the waypoint tracking objective.

The complete procedure is summarised in Algorithm 5.2. At runtime, for each waypoint k , the trained network evaluates the desired pose x_d^k , encoded as in Eq. (5.10), to produce $\hat{\mathbf{q}}_k$. This estimate is then used as the initial guess of the constrained IK solver, which refines the solution in order to satisfy the waypoint objective and the bound constraints. The refined configuration $\mathbf{q}_k$ is included in the accepted sequence only if it satisfies the tolerance in Eq. (5.7); otherwise, the waypoint is discarded.

Algorithm 5.2 Hybrid online trajectory generation, NN initialisation + constrained refinement

Input: desired waypoints $\{x_d^k\}_{k=1}^N$; trained network f_θ ; bounds of Section 5.2; tolerance ε_p , defined in Eq. (5.7).

Output: refined configuration sequence $\{\mathbf{q}_k\}$ for the accepted waypoints.

1. For $k = 1, \dots, N$:
 - (a) Evaluate the desired pose vector x_d^k .
 - (b) Compute network prediction: $f_\theta(x_d^k) \rightarrow \hat{\mathbf{q}}_k$.
 - (c) Solve the constrained IK problem Eq. (5.6) using $\hat{\mathbf{q}}_k$ as the initial guess, and obtain the refined solution $\mathbf{q}_k$.
 - (d) Compute the end-effector position $\mathbf{p}(\mathbf{q}_k)$ via forward kinematics and evaluate the waypoint error $\|\mathbf{p}(\mathbf{q}_k) - \mathbf{p}_d^k\|$.

(e) If Eq. (5.7) is satisfied, store $\mathbf{q}_k$. Otherwise, discard waypoint k .

The neural network provides an initial configuration estimate close to a feasible solution, which helps the solver converge faster. The optimisation stage then refines this estimate while enforcing the physical constraints and waypoint accuracy.

The performance of the hybrid pipeline is evaluated in the next section using the 16 trajectory–orientation combinations. For illustration, two representative trajectories, circular and helical, are discussed in detail.

5.7 Simulation Results and Performance Evaluation

This section reports the simulation results of the hybrid online pipeline introduced in Section 5.6. The evaluation follows the same trajectory–orientation combinations used for dataset generation. For illustration, trajectory tracking is presented for two representative cases: a circular path and a helical path, Figures 5.2 and 5.3. All reported statistics are computed over accepted waypoints only, using the acceptance criterion defined in Eq. (5.7).

Representative tracking results are shown in Figure 5.2, circular, and Figure 5.3, helical, under the Fixed Tilt around Y constraint. For each trajectory, two hybrid variants are reported: HybMLP and HybDNN, corresponding to MLP- and DNN-based initialisation, respectively, see Table 5.1. In both cases, the hybrid trajectories, red circle, closely follow the desired paths, blue path. The third subplot visualizes the spatial robot configuration together with the executed end-effector trajectory, where the colour gradient encodes progression along the path, providing an intuitive representation of continuous motion execution.

The key objective is to quantify the benefit of the hybrid structure relative to a purely learning-based estimate. At each waypoint, the network provides an initial configuration $\hat{\mathbf{q}}_k$, and the constrained refinement stage solves the waypoint-wise optimisation Eq. (5.6) to obtain the accepted configuration $\mathbf{q}_k$. This refinement becomes particularly important in regions of higher curvature or stricter orientation requirements, where NN-only predictions may deviate from feasible configurations.

A quantitative summary of the computational performance obtained from the simulation-based evaluation is reported in Table 5.2. Pure neural inference is extremely fast, requiring only 0.17ms for the MLP and 0.23ms for the DNN per sample. In contrast, the hybrid models include an additional constrained optimisation stage, increasing the per-waypoint computation time to approximately 21.10ms for HybMLP and 17.86ms for HybDNN. Despite this additional computational cost, the hybrid approach significantly reduces the prediction error, lowering the validation RMSE from 10.42 to 5.85 in the DNN-based model.

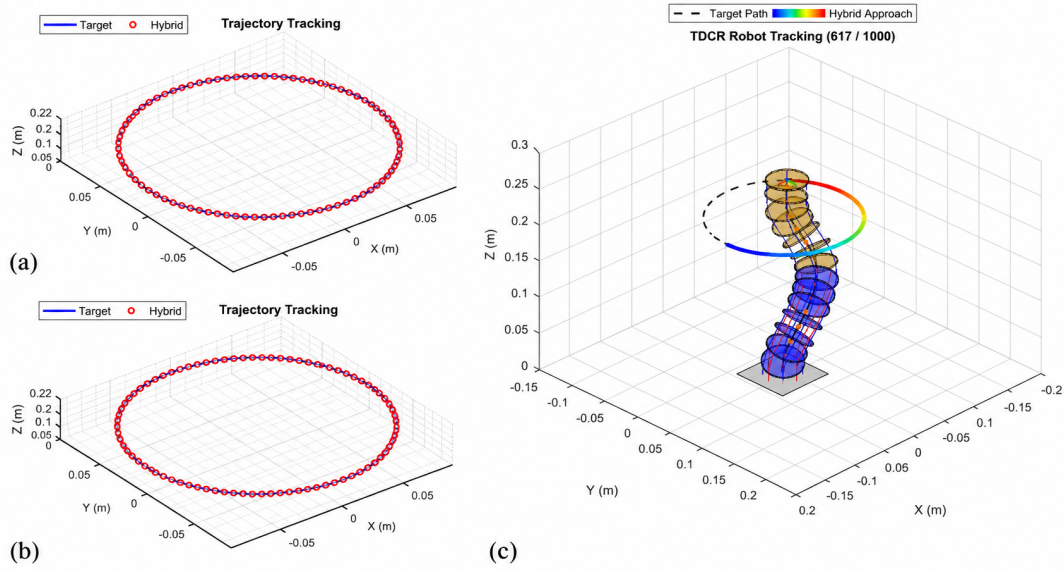


Figure 5.2 Circular trajectory tracking using the hybrid inverse kinematics framework. (a) HybMLP versus target path. (b) HybDNN versus target path. (c) Three-dimensional TDCR configuration at sample 617 out of 1000 along the trajectory. The dashed black curve denotes the target path, while the color-coded curve represents the HybDNN trajectory, with colour encoding progression along the path.

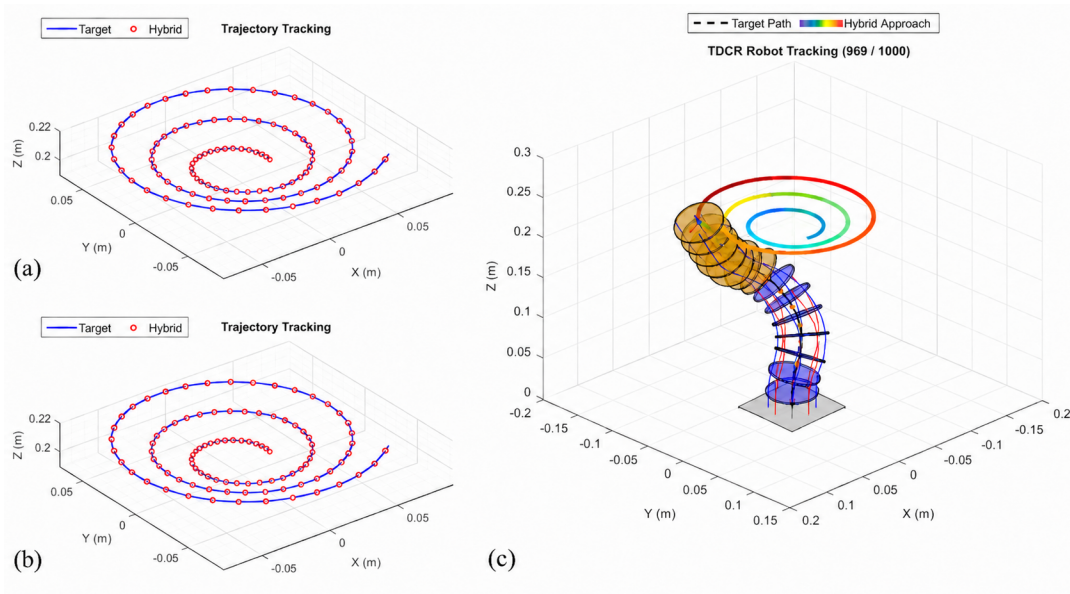


Figure 5.3 Helical trajectory tracking using the hybrid inverse kinematics framework. (a) HybMLP versus target path. (b) HybDNN versus target path. (c) Three-dimensional TDCR configuration at sample 969 out of 1000 along the trajectory. The dashed black curve denotes the target path, while the color-coded curve represents the HybDNN trajectory.

These results indicate that the optimisation refinement stage is responsible for the final improvement in accuracy.

Table 5.2 Computational performance of neural and hybrid inverse-kinematics models under the Lock Y-axis constraint ($N = 1000$ samples).

Model	Train Time (s, 1 epoch)	Inference Time (ms/sample)	Val RMSE	Num Params
MLP	0.372	0.17	9.89	7,722
DNN	0.440	0.23	10.42	29,162
HybMLP	N/A	21.10	6.52	7,722
HybDNN	N/A	17.86	5.85	29,162

Representative tendon-length variation profiles for the circular trajectory are shown in Figures 5.4 and 5.5 for Segment 1 and Segment 2, respectively. The tendon variations are plotted over the waypoint index and exhibit bounded and coordinated behaviour among antagonistic tendon pairs. This behaviour is important in practice, since abrupt tendon commands could lead to undesirable actuation discontinuities. Segment 2 shows slightly larger variations than Segment 1, which is consistent with its distal role in shaping the end-effector motion. The same analysis was performed for the remaining trajectories and orientation constraints; their tendon profiles showed similar bounded behaviour and are therefore omitted for brevity.

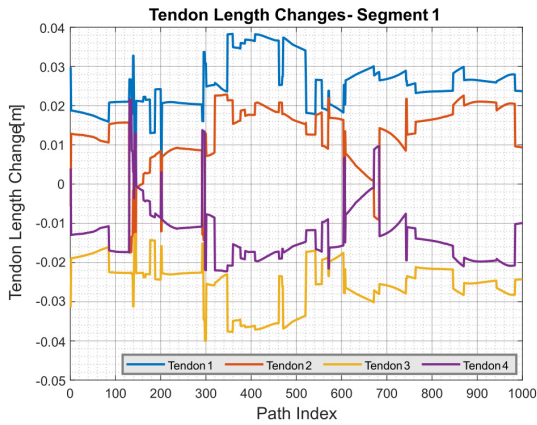


Figure 5.4 Tendon length variations in Segment 1 during circular trajectory execution, four tendons plotted over waypoint index.

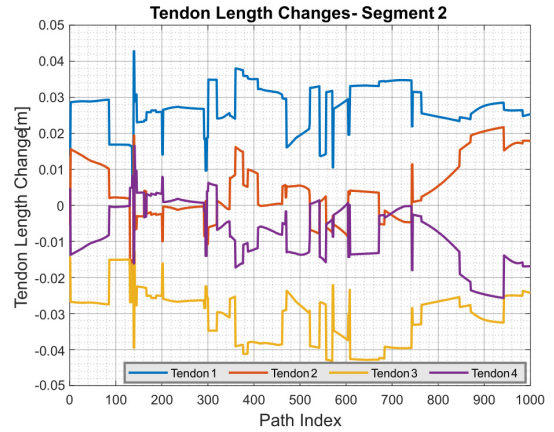


Figure 5.5 Tendon length variations in Segment 2 during circular trajectory execution, four tendons plotted over waypoint index.

These tendon-length plots provide an actuator-level view of the trajectories generated in Sections 5.5 and 5.6, by expressing the configuration sequence $\{\mathbf{q}_k\}$ as tendon commands. They also make explicit how the burden of motion execution is distributed across tendons and segments along the waypoint index.

In the next part of this chapter, the focus shifts from waypoint-by-waypoint online computation to an offline capability-map formulation, aiming to reduce runtime computation while preserving feasibility-aware execution.

5.8 Motivation for Offline Capability Mapping

The previous sections showed that the hybrid trajectory-generation pipeline improves task-space execution compared with a purely learning-based inverse-kinematics approximation. The neural network provides a fast initial estimate of the vertebra-level configuration, while the constrained refinement step improves waypoint accuracy under the adopted parameter bounds. The simulation results also showed that the corresponding tendon-length profiles remain bounded and coordinated along the trajectory.

At the same time, the main computational difficulty is not removed. Although learning improves initialisation, the final configuration at each waypoint still has to be obtained through constrained optimisation. As a result, the computational effort accumulates along the path. For short trajectories this remains manageable, but for longer paths, finer discretisations, or stricter orientation requirements, repeated refinement becomes increasingly expensive. In this sense, the hybrid formulation reduces the burden of online inverse kinematics, but does not eliminate it.

This limitation is consistent with the broader literature on continuum and tendon-driven continuum robots. These systems are attractive in confined and contact-sensitive environments because of their compliance and geometric adaptability [47–50]. At the same time, planning remains difficult because the mapping from actuation space to task space is nonlinear, redundant, and strongly constrained [48, 49]. Mechanics-based formulations provide a more faithful description of robot behaviour, but are generally too expensive for repeated online use. Geometric PCC models, including the one adopted in this thesis, are more suitable for large-scale sampling and optimisation-based inverse kinematics, but even with this simplified representation, repeated waypoint-wise refinement still limits runtime scalability [49, 51–53].

A further issue is that the numerical behaviour of inverse kinematics is not uniform across the workspace. Near high-curvature regions, near the admissible parameter bounds, or close to workspace limits, small configuration changes may produce large variations in the end-effector pose. Under such conditions, solver sensitivity increases and runtime behaviour becomes less predictable from one waypoint to the next. For a hyper-redundant robot, this makes a purely online strategy less attractive when consistent execution is required.

For this reason, the remainder of this chapter adopts a different strategy in which the expensive inverse-kinematics computations are shifted offline. Rather than solving a constrained problem repeatedly during execution, the task space is explored in advance and organised in a structured representation that can later be queried at runtime. This idea is related to capability-map methods used in rigid-link robotics [54–58], where workspace regions are associated with precomputed performance information. For the tendon-driven continuum robot studied here, however, geometric reachability alone is not sufficient. A target pose is useful only if it can be associated with at least one configuration that is compatible with the adopted PCC parametrisation, the imposed parameter bounds, and the prototype-level tendon-routing and actuation-mapping model.

Moreover, multiple feasible configurations may coexist within the same workspace neighbourhood while exhibiting different pose residuals, internal shapes, and local kinematic properties. Therefore, the offline representation should not only identify reachable regions, but should also preserve feasible configuration alternatives for later selection. In this thesis, the CAP is therefore treated as a database-driven representation that supports voxel-level configuration retrieval rather than repeated online inverse-kinematics solving for every queried target.

The next section introduces this capability-map representation for the studied two-segment TDCR. The task space is discretised into voxelised spatial regions, each region is paired with sampled tool orientations, and the corresponding inverse-kinematics problems are solved offline. The retained solutions are first stored together with the associated configuration vectors and pose residuals, and are later enriched with dexterity descriptors so that runtime execution can be based on selection from precomputed feasible alternatives rather than repeated online inverse kinematics.

5.9 Capability-Map Database Representation of the Studied TDCR

In the present work, the capability map (CAP) is defined as an offline database that organises feasible task-space targets together with the corresponding robot configurations obtained under the adopted vertebra-wise PCC model and bounded parametrisation. As discussed in the previous section, the main distinction from conventional rigid-link capability maps is that geometric reachability alone is not sufficient for the studied TDCR. A target pose is relevant only if it admits at least one configuration that satisfies the adopted parameter bounds and

remains compatible with the prototype-level tendon-routing and actuation-mapping model. The CAP is therefore constructed as a feasibility-aware task-space database from the outset.

Following the kinematic representation introduced in Chapter 4, the robot configuration is described by the stacked vertebra-wise PCC variables

$$\mathbf{q}_i = \begin{bmatrix} \tilde{\kappa}_i & \varphi_i & \tau_i \end{bmatrix}^T, \quad \mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \cdots & \mathbf{q}_{14}^T \end{bmatrix}^T, \quad (5.12)$$

with $\tilde{\kappa}_i \in [-25, 25] \text{ m}^{-1}$, $\varphi_i \in [0, 2\pi] \text{ rad}$, $\tau_i \in [-6, 6] \text{ rad/m}$.

Here, φ_i and τ_i denote bending-plane angle and twist rate of the i th vertebra-level element. The variable $\tilde{\kappa}_i$ is used as a signed curvature parameter in the numerical implementation, while the physical curvature magnitude is $\kappa_i = |\tilde{\kappa}_i|$, as defined in Chapter 4. This signed representation allows the optimisation to explore bending on both sides of the straight configuration without introducing separate sign cases.

The task space is discretised into voxelised spatial neighbourhoods. Each voxel centre $\mathbf{p}_v$ is paired with a finite set of sampled tool orientations $\mathbf{R}_{v,s}$, where v denotes the voxel index and s denotes the sampled orientation index assigned to that voxel. A CAP target is therefore defined by a voxel–orientation pair $(\mathbf{p}_v, \mathbf{R}_{v,s})$. For each pair, the inverse-kinematics problem is solved offline under the same PCC model and parameter bounds. If the obtained solution satisfies the prescribed pose-accuracy thresholds and remains within the admissible variable limits, it is retained as a valid database entry. Each retained entry stores the target pose, the solver-found pose, the corresponding configuration vector, and the residual pose errors.

An important property of the adopted representation is that redundancy is preserved rather than collapsed into a single solution. For a given voxel, multiple retained entries may exist, possibly corresponding to different sampled tool orientations, different internal PCC shapes, and different residual values. These entries may also exhibit different local kinematic properties. The CAP is therefore constructed as a set-valued representation, in which a single task-space neighbourhood can be associated with several feasible alternatives. This property is essential for the selection stage developed later in the chapter, where the final runtime choice is based not only on pose feasibility but also on the local kinematic quality of the retained configuration.

It is also important to clarify the level at which the CAP is queried. In this thesis, the runtime stage is formulated as a voxel-level configuration retrieval problem, rather than as an exact inverse-kinematics solver for an arbitrary prescribed pose. Therefore, when a voxel is queried, the database entries associated with that local neighbourhood are retrieved and ranked using the stored residuals and the dexterity descriptors computed offline. This allows

the online stage to select a representative best-compromise configuration without solving a new inverse-kinematics problem.

On this basis, the following sections describe how the CAP database is constructed and then used for configuration selection. The construction process first defines the region of interest, voxelisation, orientation sampling, PSO-based inverse kinematics, actuation mapping, and acceptance filtering. The retained entries are then enriched with offline dexterity descriptors, which are later used together with the stored pose residuals to select a best-compromise configuration within a queried voxel.

5.10 Offline Construction of the Capability Database

The capability-map representation defined in the previous section is constructed offline over a selected region of the workspace of the two-segment TDCR. To motivate this choice, the global reachable workspace is first considered. Figure 5.6(a) shows the sampled reachable task-space envelope obtained by forward-kinematic sampling over the admissible range of the vertebra-wise PCC parameters, providing an indicative view of the positions reachable by the robot tip.

Based on this envelope, the CAP is not constructed over the entire reachable workspace. Instead, a distal region of interest (ROI) is selected around the expected operating area of the tool tip, as shown in Figure 5.6(b) and Figure 5.6(c). This focused choice allows dense voxel–orientation sampling and sufficient database resolution within a controlled workspace region, while keeping the offline computational cost and database size manageable.

To obtain a compact and overlap-free neighbourhood representation, each voxel centre is associated with a spherical neighbourhood of fixed radius $r_{\text{sphere}} = 5 \text{ mm}$. Accordingly, voxel centres are placed on a uniform grid with spacing $2r_{\text{sphere}}$ over $x, y \in [-25, 25] \text{ mm}$ and $z \in [L - 20, L + 20] \text{ mm}$, where $L = 265.26 \text{ mm}$ is the total length of the two segments. This discretisation provides $N_v = 180$ voxels in the selected ROI.

Let $\mathbf{p}_v$ denote the centre of voxel v . To represent pose-level rather than position-only capability, each voxel is paired with a finite set of sampled tool-axis directions, from which full target orientations are subsequently constructed. Throughout CAP construction, a consistent z-down tool convention is enforced. This means that the sampled target orientations are restricted to downward tool-axis directions, in agreement with the robot-frame convention adopted in this chapter.

For each voxel centre $\mathbf{p}_v$, candidate tool-axis directions are generated by regular angular discretisation of the lower hemisphere. This lower-hemisphere sampling for one representa-

tive voxel is illustrated in Figure 5.6(d). In a local spherical parametrisation, α denotes the azimuth angle around the local z -axis, while β denotes the inclination angle measured from the positive local z -axis. The azimuth angle is sampled with $N_\alpha = 30$ uniformly spaced values over $[0, 2\pi]$, while the inclination angle is sampled with $N_\beta = 10$ uniformly spaced values over $[\pi/2, \pi)$. This gives $N_{\text{sample}} = N_\alpha N_\beta = 300$ candidate tool-axis directions per voxel. Figure 5.6(e) shows a bottom-view projection of the 300 distributed candidates, illustrating the regular angular coverage of the sampled directions.

Each sampled direction is converted into a full target orientation matrix $\mathbf{R}_{v,s}$ using the frame-construction rule adopted for the z -down convention. A complete CAP target is therefore defined by the voxel orientation pair $(\mathbf{p}_v, \mathbf{R}_{v,s})$, where s indexes within voxel v . With 180 voxels and 300 sampled directions per voxel, the full offline target set contains 54,000 voxel–orientation samples for inverse-kinematics evaluation.

5.10.1 PSO-Based Inverse Kinematics and Acceptance Filtering

For each target pose $(\mathbf{p}_v, \mathbf{R}_{v,s})$ generated over the discretized ROI, inverse kinematics is solved offline under the same bounded PCC parameterization adopted in the previous sections. Using the stacked configuration vector $\mathbf{q}$ together with the bounds on $\tilde{\kappa}_i$, ϕ_i , and τ_i defined above, the offline IK problem is formulated as

$$\mathbf{q}^* = \arg \min_{\mathbf{q} \in \mathbb{R}^{3N}} \mathcal{J}_{\text{IK}}(\mathbf{q}), \quad (5.13)$$

where the objective combines translational error and geodesic orientation error,

$$\mathcal{J}_{\text{IK}}(\mathbf{q}) = \|\mathbf{p}(\mathbf{q}) - \mathbf{p}_v\|_2 + \lambda_{\text{ang}} \text{dist}_{SO(3)}(\mathbf{R}(\mathbf{q}), \mathbf{R}_{v,s}). \quad (5.14)$$

The orientation term is computed through the geodesic distance on $SO(3)$,

$$\text{dist}_{SO(3)}(\mathbf{R}_1, \mathbf{R}_2) = \arccos \left(\frac{\text{trace}(\mathbf{R}_1^T \mathbf{R}_2) - 1}{2} \right). \quad (5.15)$$

The angular-to-linear scaling factor is taken as

$$\lambda_{\text{ang}} = 0.25L, \quad L = \sum_{j=1}^N \delta_j. \quad (5.16)$$

Here, $\mathbf{p}(\mathbf{q})$ and $\mathbf{R}(\mathbf{q})$ are obtained from the forward kinematics of the discretized TDCR. The factor λ_{ang} provides the adopted scaling between orientation and position errors, so that the rotational term remains comparable to the translational term in the combined objective.

The optimisation is solved using Particle Swarm Optimisation (PSO). This choice is motivated by the nonlinear and nonconvex structure of the IK problem. With $3N = 42$

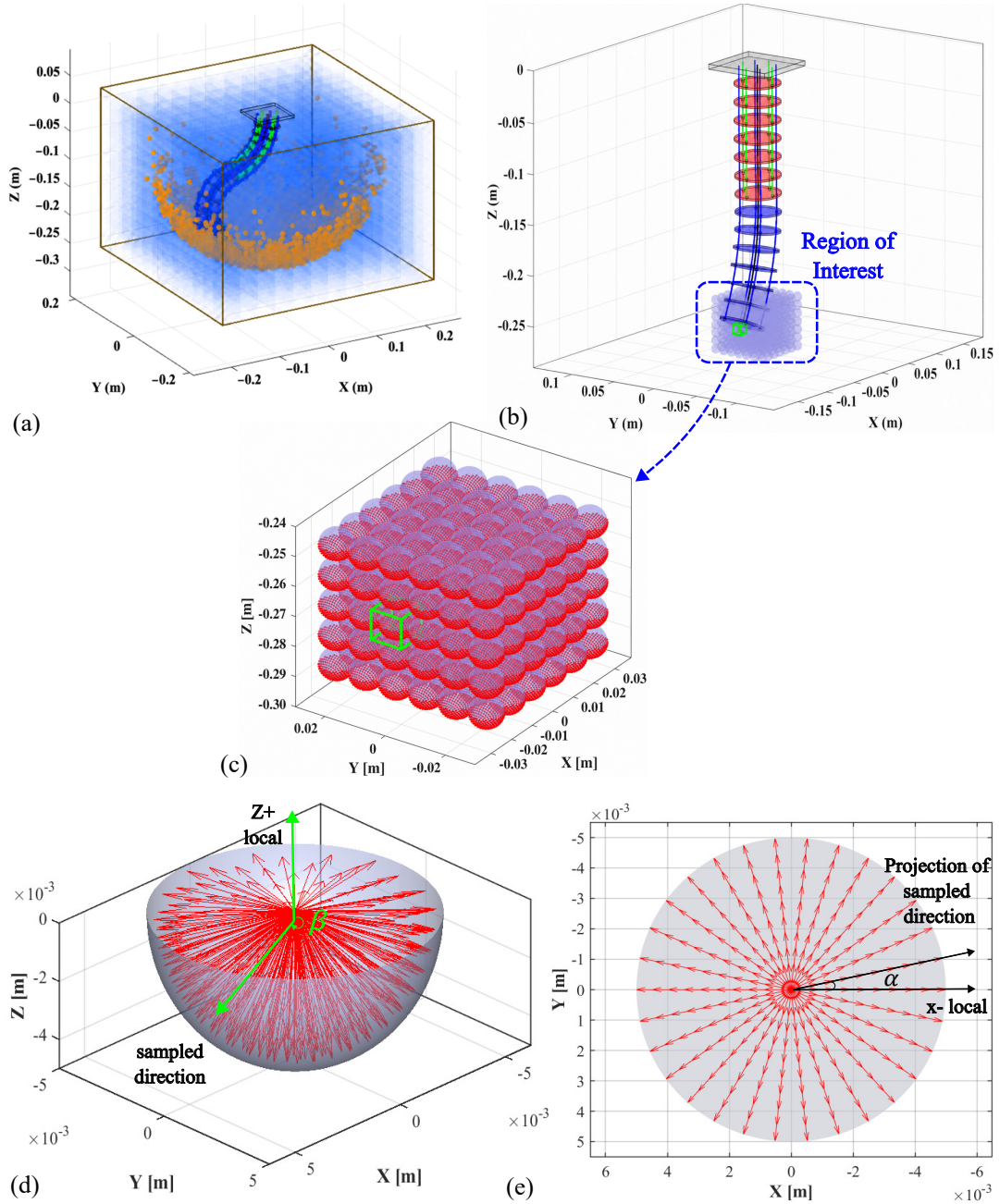


Figure 5.6 Workspace and CAP sampling strategy: (a) sampled reachable workspace of the two-segment TDCR obtained through forward-kinematic sampling; (b) TDCR model with the selected distal region of interest; (c) isolated view of the selected ROI and its voxelised representation; (d) lower-hemisphere sampling of candidate tool-axis directions for one representative voxel; and (e) bottom-view projection of the 300 distributed candidate directions.

optimisation variables and a high-dimensional PCC forward-kinematic map, local optimisation may become sensitive to the initial condition, especially near workspace boundaries, high-curvature regions, or admissible parameter limits. Since the CAP is generated offline,

a broader stochastic search can be used without affecting the runtime query stage. In the present implementation, the solver uses the configuration bounds defined above for $\tilde{\kappa}_i$, φ_i , and τ_i , a swarm size of 50 particles, a maximum of 150 iterations, and a function tolerance of 10^{-6} .

For acceptance checking, the translational and rotational residuals of the solver-found configuration are evaluated as

$$e_p = \|\mathbf{p}(\mathbf{q}^*) - \mathbf{p}_v\|_2, \quad e_R = \text{dist}_{SO(3)}(\mathbf{R}(\mathbf{q}^*), \mathbf{R}_{v,s}). \quad (5.17)$$

A sample is retained in the CAP database only if both thresholds are satisfied simultaneously,

$$e_p \leq 5 \text{ mm}, \quad e_R \leq 10^\circ. \quad (5.18)$$

Voxel centres that fail a conservative reachability check based on the available backbone length are skipped before the full optimisation step, reducing unnecessary solver evaluations. For the remaining targets, the PSO solver returns a candidate configuration together with its associated residuals. Samples satisfying Eq. (5.18) are stored as feasible CAP entries, whereas unsuccessful and skipped cases are recorded separately for later analysis. The retained database therefore contains only those voxel–orientation samples that satisfy the adopted pose-accuracy thresholds under the same PCC model and parameter bounds used throughout this chapter.

5.10.2 Actuation Mapping

To relate each solver-found configuration to the physical robot, the corresponding tendon-length variations are computed and converted into motor commands. For each retained sample, the deformed length of tendon $c \in \{1, \dots, 8\}$ is evaluated from the tendon guide points distributed along the discretised backbone,

$$\ell_c(\mathbf{q}) = \sum_{j=1}^N \|\mathbf{p}_{c,i}(\mathbf{q}) - \mathbf{p}_{c,i+1}(\mathbf{q})\|_2, \quad \Delta\ell_c(\mathbf{q}) = \ell_c(\mathbf{q}) - \ell_c^{\text{rest}}. \quad (5.19)$$

Here, ℓ_c^{rest} is the tendon rest length in the reference straight configuration. In the present prototype, this reference depends on the segment to which the tendon belongs: for Segment 1, $\ell_c^{\text{rest}} = 135 \text{ mm}$, and for Segment 2, $\ell_c^{\text{rest}} = 275 \text{ mm}$. The quantities $\Delta\ell_c(\mathbf{q})$ represent the tendon displacements associated with the corresponding PCC configuration.

For execution on the motor-driven prototype, these linear displacements are converted into motor encoder counts. Given the pulley radius $r_{\text{pulley}} = 10.5 \text{ mm}$ and the motor resolution

$N_{\text{enc}} = 4096$ counts/rev, the command associated with tendon c is computed as

$$u_c = \text{round} \left(\frac{\Delta \ell_c(\mathbf{q})}{r_{\text{pulley}}} \frac{N_{\text{enc}}}{2\pi} \right). \quad (5.20)$$

Since each motor shaft drives an antagonistic tendon pair, the final shaft command is obtained by applying equal and opposite tendon displacements within the corresponding pair. Thus, each retained CAP entry is associated with a feasible PCC configuration and the corresponding prototype-level actuation command under the adopted tendon-routing and pulley-mapping model.

For each retained sample, the CAP database stores the voxel identifier, the sampled orientation identifier, the target pose $(\mathbf{p}_v, \mathbf{R}_{v,s})$, the optimized configuration vector $\mathbf{q}^*$, the achieved pose, the associated residual errors (e_p, e_R) , and the corresponding actuation command vector.

5.11 Offline Differential Kinematics and Dexterity Evaluation

After the offline CAP database has been generated, each retained entry is further evaluated in terms of its local differential kinematic behaviour. The retained samples obtained in the previous section are already feasible with respect to the adopted PCC parametrisation, pose-accuracy thresholds, and actuation-mapping model. However, feasibility alone does not indicate whether a configuration provides favourable local motion capability. For this reason, each retained configuration is enriched offline with Jacobian-based dexterity descriptors.

For a retained configuration $\mathbf{q}$, the forward kinematics provides the end-effector transformation ${}^B\mathbf{T}_E(\mathbf{q})$, from which the position $\mathbf{p}(\mathbf{q}) \in \mathbb{R}^3$ and orientation $\mathbf{R}(\mathbf{q}) \in SO(3)$ are extracted. The local differential relationship between the configuration velocity and the end-effector twist is written as

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix}, \quad \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{v}(\mathbf{q}) \\ \boldsymbol{\omega}(\mathbf{q}) \end{bmatrix}, \quad (5.21)$$

where $\mathbf{J}_p(\mathbf{q}) \in \mathbb{R}^{3 \times 3N}$ and $\mathbf{J}_\omega(\mathbf{q}) \in \mathbb{R}^{3 \times 3N}$ denote the translational and rotational parts of the Jacobian, respectively. In this chapter, the Jacobian is not used for online inverse kinematics. Instead, it is computed offline for the retained CAP samples and used to assign local dexterity descriptors to the corresponding database entries.

For the adopted disk-wise PCC parametrisation, deriving closed-form Jacobian expressions becomes increasingly complex when all curvature, bending-plane, and twist variables

across both segments are included. The Jacobian is therefore computed numerically from the forward-kinematic map. During this differentiation step, the raw forward kinematics is used without re-applying the target z-down frame-construction rule, in order to avoid artificial discontinuities in the local linearisation.

For the n th configuration variable, with basis vector $\mathbf{e}_n$ and perturbation step δ_n , the translational Jacobian column is approximated using central finite differences as

$$\mathbf{J}_p(:,n)(\mathbf{q}) \approx \frac{\mathbf{p}(\mathbf{q} + \delta_n \mathbf{e}_n) - \mathbf{p}(\mathbf{q} - \delta_n \mathbf{e}_n)}{2\delta_n}. \quad (5.22)$$

The rotational part is computed in the tangent space of $SO(3)$. Here, $\log(\cdot)$ denotes the matrix logarithm on $SO(3)$, and $\text{vee}(\cdot)$ maps a skew-symmetric matrix in $\mathfrak{so}(3)$ to its vector representation in $\mathbb{R}^3$. Defining

$$\xi_n^\pm = \text{vee}(\log(\mathbf{R}(\mathbf{q})^T \mathbf{R}(\mathbf{q} \pm \delta_n \mathbf{e}_n))), \quad (5.23)$$

the corresponding rotational Jacobian column is approximated as

$$\mathbf{J}_\omega(:,n)(\mathbf{q}) \approx \frac{\xi_n^+ - \xi_n^-}{2\delta_n}. \quad (5.24)$$

This numerical procedure provides a local differential approximation of the PCC kinematics for each retained CAP entry. Since the computation is performed offline, it does not affect the runtime database query stage.

Once the Jacobian has been computed, each feasible CAP sample is assigned a scalar dexterity value. In the present framework, the main descriptor is the 6D manipulability measure, because the subsequent selection stage must account for both translational and rotational motion capability when ranking feasible configurations within a queried voxel. To make these two contributions commensurate, the same angular-to-linear scaling factor λ_{ang} introduced in the IK objective is retained. This keeps the position–orientation weighting convention consistent across offline IK generation, residual evaluation, and dexterity characterisation. The scaled Jacobian is therefore defined as

$$\mathbf{J}_6(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \lambda_{\text{ang}} \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \quad (5.25)$$

and the corresponding 6D manipulability descriptor is computed as

$$w_{6D}(\mathbf{q}) = \sqrt{\det(\mathbf{J}_6(\mathbf{q}) \mathbf{J}_6(\mathbf{q})^T)}. \quad (5.26)$$

This quantity is used as the dexterity term in the database-driven selection stage. In addition, a positional manipulability descriptor is computed offline for descriptive analysis

of the CAP database,

$$w_p(\mathbf{q}) = \sqrt{\det(\mathbf{J}_p(\mathbf{q})\mathbf{J}_p(\mathbf{q})^T)}. \quad (5.27)$$

While w_p provides useful insight into the translational contribution to local dexterity, it is not adopted as the main selection objective. The reason is that the CAP selection problem is pose-level rather than position-only; therefore, the selected descriptor should account for both translational and rotational motion capability.

Each retained CAP entry is therefore characterised not only by its target pose, optimised configuration, achieved pose, residual errors, and actuation command, but also by local dexterity descriptors computed offline from the corresponding Jacobian.

5.12 Database-Driven Bi-Objective Configuration Selection

Once the CAP database has been enriched with dexterity descriptors, the remaining step is to define how a single configuration is selected for a queried voxel. The proposed selector is not formulated as a pose-conditioned inverse-kinematics solver for arbitrary target poses. Instead, it retrieves the precomputed entries associated with the queried voxel and selects one representative configuration from this finite candidate set. Since entries generated for different sampled tool-axis directions within the same voxel may be considered together, the runtime query is defined at the voxel level rather than at the exact-pose level.

Let $\mathcal{D}$ denote the offline CAP database. For a queried voxel v , the candidate set is defined as

$$\mathcal{S}_v = \{i \mid \text{voxelID}(i) = v\}. \quad (5.28)$$

Each candidate $i \in \mathcal{S}_v$ is a retained CAP entry whose original IK query shared the same voxel centre $\mathbf{p}_v$, but may have been generated for a different sampled tool-axis direction. Thus, $\mathcal{S}_v$ contains orientation-diverse feasible entries within the same local task-space neighbourhood.

The selection problem is defined by two objectives: stored pose quality and local dexterity. It is written as

$$\min_{i \in \mathcal{S}_v} \begin{bmatrix} f_1(i) \\ f_2(i) \end{bmatrix}. \quad (5.29)$$

The first objective evaluates the stored pose quality of each candidate through a combined residual measure,

$$f_1(i) = \sqrt{e_p(i)^2 + (\lambda_{\text{angle}} e_R(i))^2}, \quad (5.30)$$

where $e_p(i)$ and $e_R(i)$ are the stored position and orientation residuals of candidate i , computed with respect to the target pose of that specific CAP entry. The same scaling factor λ_{ang} used in the offline IK stage is retained here to keep the position–orientation weighting convention consistent.

The second objective promotes configurations with higher local motion capability by minimizing the inverse of the precomputed 6D manipulability,

$$f_2(i) = \frac{1}{w_{6D}(i) + \varepsilon}, \quad (5.31)$$

where w_{6D} is the offline dexterity descriptor defined in Eq. (5.26), and $\varepsilon > 0$ is a small constant used to avoid numerical problems near singular configurations. Therefore, smaller values of f_1 indicate better stored pose accuracy, while smaller values of f_2 indicate higher local manipulability.

Since the two objectives may conflict, the selection is first restricted to the non-dominated subset of $\mathcal{S}_v$. A candidate A dominates a candidate B if

$$(f_1(A) \leq f_1(B) \wedge f_2(A) \leq f_2(B)) \wedge (f_1(A) < f_1(B) \vee f_2(A) < f_2(B)). \quad (5.32)$$

The Pareto set $\mathcal{P}_v \subseteq \mathcal{S}_v$ is formed by all candidates that are not dominated by any other element of $\mathcal{S}_v$. This step removes clearly inferior entries while preserving the feasible trade-offs between stored pose quality and dexterity within the queried voxel.

To obtain a single selected configuration, a best-compromise solution is chosen from $\mathcal{P}_v$ using a normalized ideal-point criterion. If the Pareto set contains only one candidate, that configuration is selected directly. Otherwise, each objective is min–max normalized over $\mathcal{P}_v$ as

$$\tilde{f}_b(i) = \frac{f_b(i) - f_b^{\min}}{f_b^{\max} - f_b^{\min}}, \quad b \in \{1, 2\}, \quad (5.33)$$

where $f_b^{\min}$ and $f_b^{\max}$ denote the minimum and maximum values of f_b within the Pareto set. The final selection is then obtained by minimizing the Euclidean distance to the ideal point $(0, 0)$,

$$i^* = \arg \min_{i \in \mathcal{P}_v} \sqrt{\tilde{f}_1(i)^2 + \tilde{f}_2(i)^2}. \quad (5.34)$$

This procedure provides a deterministic best-compromise configuration for the queried voxel using only quantities already stored in the CAP database. The online stage is therefore reduced to database filtering, Pareto-set extraction, normalization, and distance evaluation. The selected configuration should be interpreted as a dexterity-aware representative entry for

the local voxel neighbourhood, rather than as the exact IK solution of a newly prescribed target pose.

5.13 Offline CAP Analysis and Configuration Selection Results

This section analyses the offline CAP database generated for the two-segment TDCR and the resulting configuration-selection behaviour. The analysis first examines the effective coverage of the discretised ROI under the adopted kinematic constraints. Among the 180 discretised voxels, 88 voxels satisfy the preliminary reachability condition, corresponding to 48.9% of the selected ROI. The remaining 92 voxels, corresponding to 51.1%, are classified as unreachable under the same constraints.

Figure 5.7 shows the resulting reachability map. Green voxel centres correspond to the reachable subset, while red voxel centres denote unreachable locations. The reachable voxels are mainly concentrated in the central and distal part of the ROI, whereas boundary regions exhibit reduced accessibility. This trend is consistent with the location of the selected ROI relative to the boundary of the reachable workspace, where fewer feasible voxel-level samples are expected under the finite bending capability of the robot.

These values describe the effective coverage of the selected local ROI and should not be interpreted as a global reachability measure of the entire TDCR workspace. The following analysis then evaluates representative PSO-based inverse-kinematics outcomes, the distribution of the offline dexterity descriptors, and the behaviour of the residual–dexterity Pareto selection within selected voxels. This is important because the CAP is used in this thesis as a configuration-level database, rather than only as a geometric reachability map.

5.13.1 IK Solver Quality

The numerical behaviour of the PSO-based IK solver is examined through representative target samples distributed across the selected ROI. Figures 5.8 and 5.9 illustrate one representative example corresponding to Sample ID 47944 in Voxel 160. Figure 5.8 compares the target frame $\mathbf{T}_{\text{target}}$ with the solver-found frame $\mathbf{T}_{\text{found}}$ inside the corresponding voxel neighbourhood, while Figure 5.9 shows the reconstructed TDCR backbone associated with the same solution.

For this sample, the solver-found configuration gives a position error of approximately 0.071 mm and an orientation error of approximately 0.030° . The reconstructed backbone

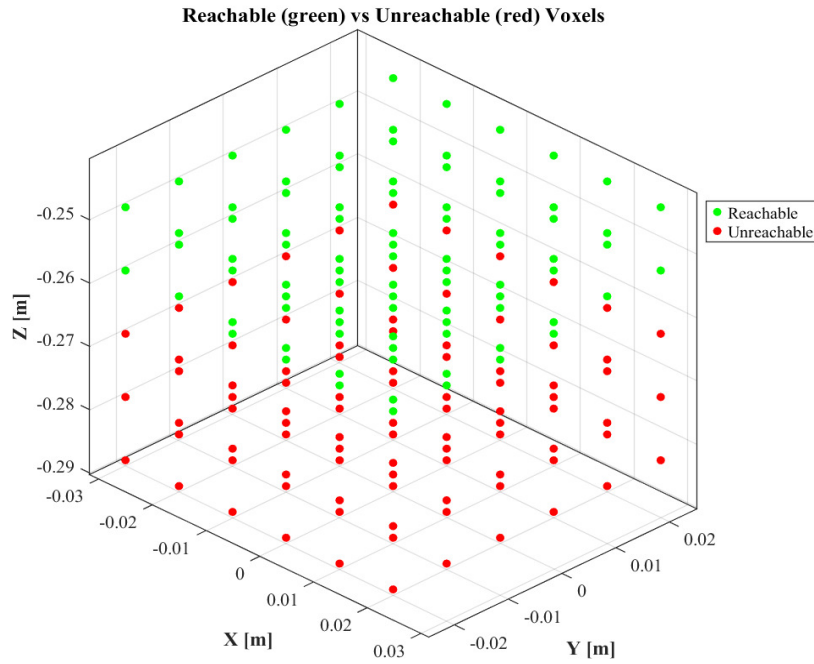


Figure 5.7 Reachability classification of the selected ROI: 88 reachable voxels, green, 48.9%, and 92 unreachable voxels, red, 51.1%, under the adopted backbone-length and curvature constraints.

remains smooth and geometrically consistent with the adopted vertebra-wise PCC model, indicating that the optimisation converges to a configuration within the prescribed kinematic bounds.

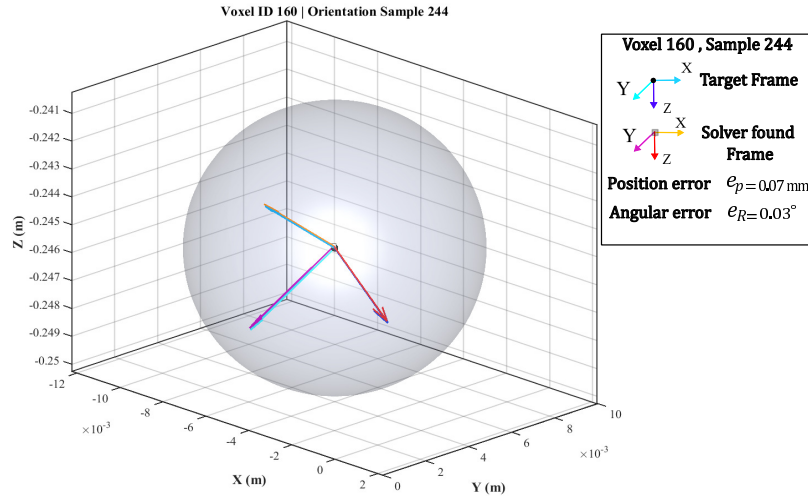


Figure 5.8 Representative voxel-level IK result showing the target frame and the solver-found pose inside the spherical neighbourhood associated with the selected voxel.

Table 5.3 reports five representative raw PSO-based IK outcomes before CAP acceptance filtering. These examples are included to illustrate the range of solver responses across

different ROI locations and should not all be interpreted as retained database entries. Only solutions satisfying both thresholds defined in Eq. (5.18), namely $e_p \leq 5$ mm and $e_R \leq 10^\circ$, are stored in the final CAP database. As shown in the table, some raw PSO outputs exceed one or both thresholds and are therefore excluded from the retained CAP entries.

Table 5.3 Representative raw IK outcomes for five sampled voxel–orientation targets before the CAP acceptance filtering.

Sample ID	Voxel ID	Orientation Sample s	Position Error $\ e_p\ $ (mm)	Orientation Error $\ e_R\ $ (deg)	Cost $\mathcal{J}_{IK}(\mathbf{q})$	Retained in CAP
7317	25	117	46.00	4.875	0.05164	No
47944	160	244	0.071	0.030	0.00010	Yes
50359	168	259	0.050	1.082	0.00130	Yes
24060	81	60	23.30	19.136	0.04544	No
20964	70	264	17.67	0.005	0.01767	No

The retained and rejected examples in Table 5.3 show that the acceptance rule depends on both translational and rotational accuracy. For instance, Sample 47944 and Sample 50359 satisfy the storage thresholds and are retained, whereas Samples 7317, 24060, and 20964 are rejected because their position error, orientation error, or both exceed the prescribed limits. This filtering step ensures that the final CAP database contains only voxel–orientation entries that meet the adopted pose-accuracy requirements.

5.13.2 Distribution of the Adopted Dexterity Metric

The offline Jacobian evaluation reveals substantial variability in dexterity across the feasible CAP entries. In this chapter, the 6D manipulability measure w_{6D} is adopted as the main dexterity descriptor because it accounts for both translational and rotational motion capability. The auxiliary positional manipulability measure w_p is also computed to provide additional insight into the translational contribution to local dexterity.

Figure 5.10 shows the global distributions of w_{6D} and w_p over all feasible CAP samples. Both distributions are highly non-uniform, with a small number of configurations exhibiting much larger manipulability values than the majority of feasible entries. This confirms that geometric reachability alone does not imply comparable local motion capability across the database.

Figure 5.11 presents the same feasible CAP samples in the (w_p, w_{6D}) plane using logarithmic axes. The resulting trend indicates a positive association between positional manipulability and overall 6D dexterity: configurations with higher translational capability tend, in general, to also provide higher w_{6D} . However, the scatter around this trend confirms that w_p cannot be used as a direct substitute for w_{6D} , since the adopted 6D metric also includes the rotational contribution required for full pose-level configuration selection.

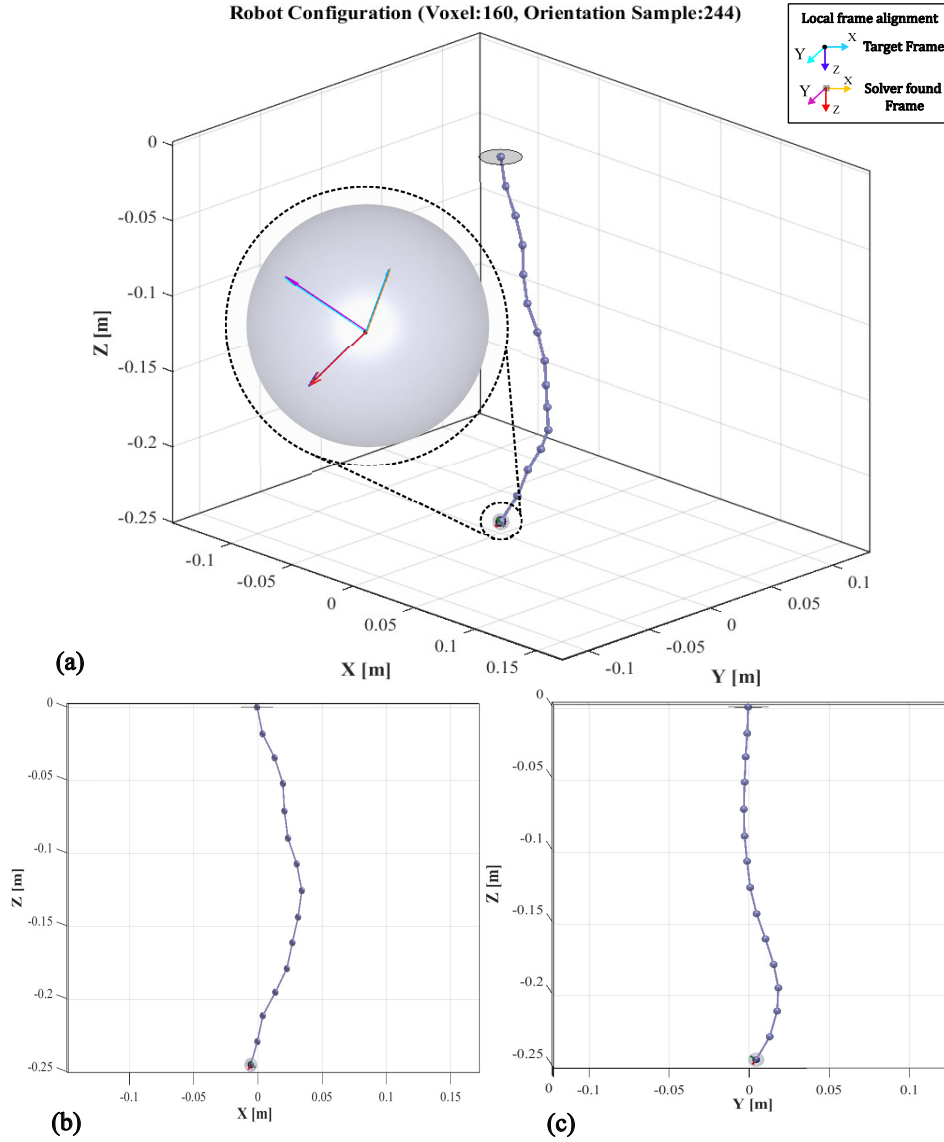


Figure 5.9 Representative solver-found TDCR configuration. (a) 3D reconstructed backbone with the selected voxel and local frame. (b) $x-z$ projection. (c) $y-z$ projection.

Dexterity varies not only across the workspace but also among the feasible configurations stored within individual voxels. Figure 5.12 examines this variability at the voxel level. Each boxplot summarises the distribution of w_{6D} over the feasible CAP entries associated with a selected voxel. The wide spreads and outliers observed in several voxels show that configurations available within the same local task-space neighbourhood may differ substantially in local kinematic quality. This confirms that configuration selection within a voxel is nontrivial and cannot be based on pose feasibility alone.

Figure 5.13 ranks workspace voxels according to their mean w_{6D} values. The comparison between the ten voxels with the highest and lowest mean dexterity shows that the local

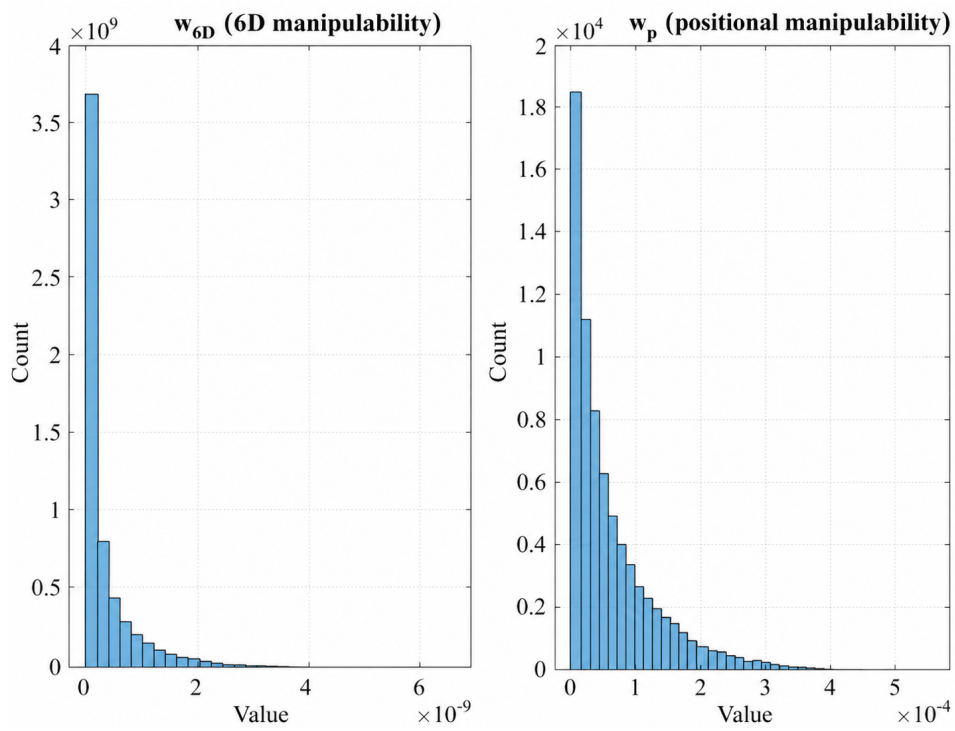


Figure 5.10 Global distributions of the 6D manipulability measure w_{6D} and the positional manipulability measure w_p over all feasible CAP entries.

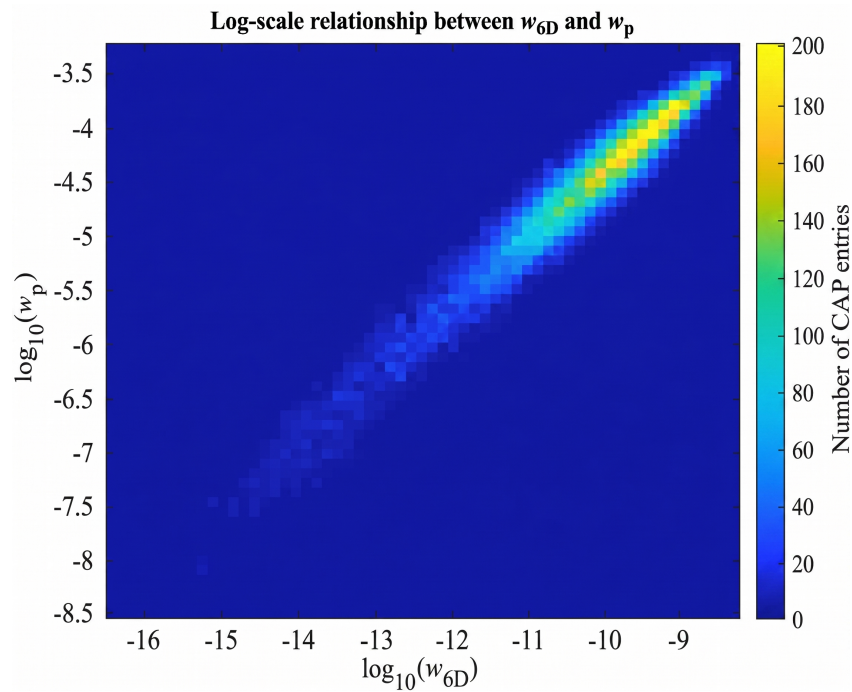


Figure 5.11 Log-scale relationship between the 6D manipulability measure w_{6D} and the positional manipulability measure w_p over all feasible CAP entries.

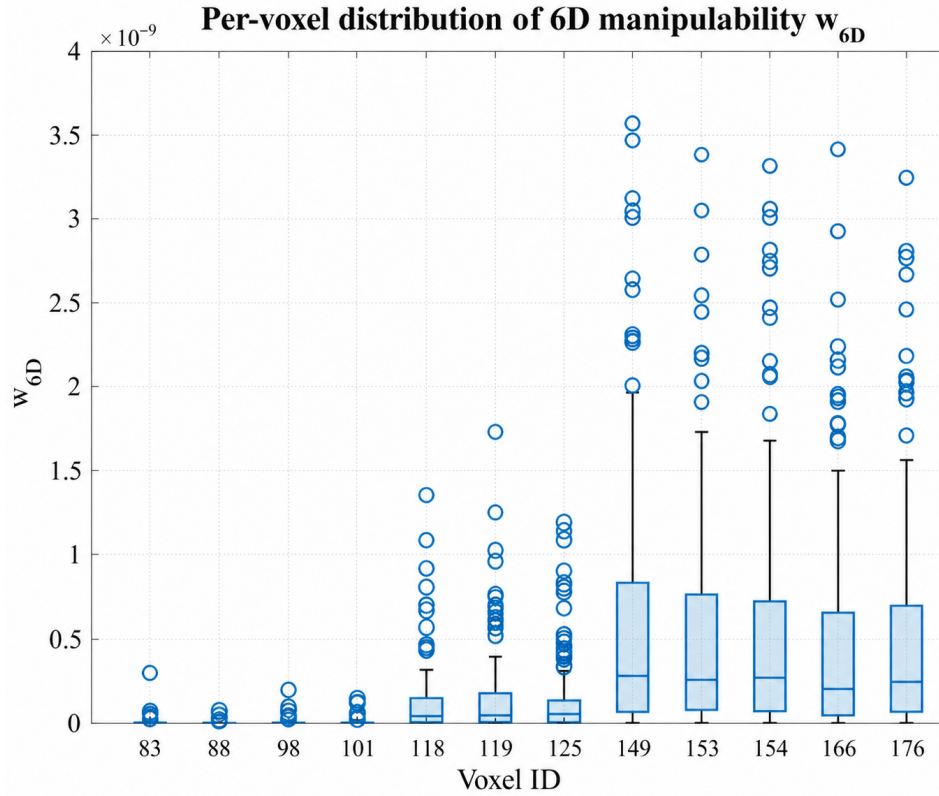


Figure 5.12 Distribution of the 6D manipulability measure w_{6D} over feasible CAP entries within selected workspace voxels.

kinematic quality of feasible configurations depends strongly on the workspace region. This observation further supports the use of a dexterity-aware selection strategy rather than a purely reachability-based or residual-only criterion.

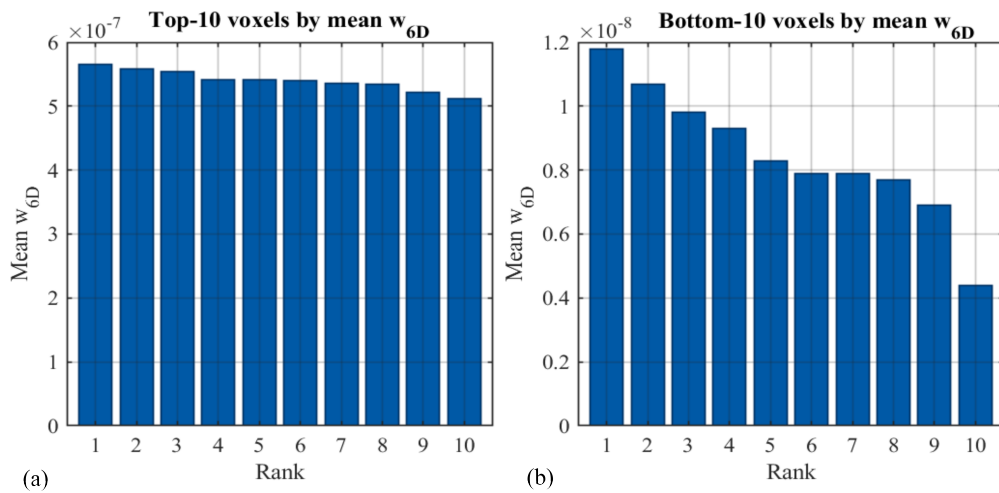


Figure 5.13 Ranking of workspace voxels based on the mean 6D manipulability measure w_{6D} . (a) Ten voxels with the highest mean w_{6D} . (b) Ten voxels with the lowest mean w_{6D} .

5.13.3 Bi-Objective Selection Analysis

The effect of the database-driven selection strategy is illustrated through representative voxel-level examples. For each queried voxel, the feasible CAP entries are evaluated using the two precomputed objectives introduced in Section 5.12: the combined pose residual f_1 and the inverse 6D manipulability f_2 . Since both quantities are already stored in the CAP database, the selection process is carried out through database filtering, Pareto-set extraction, and ideal-point ranking, without solving a new inverse-kinematics problem online.

It is important to clarify the interpretation of the bi-objective plots. Each point represents one retained CAP entry within the queried voxel, not a repeated solution for the same exact pose. Since entries within $\mathcal{S}_v$ may originate from IK queries for different sampled tool-axis directions, f_1 is computed with respect to the stored target pose of each individual entry rather than a common reference pose. The resulting Pareto set therefore represents the non-dominated residual–dexterity trade-offs available within the queried voxel.

Figures 5.14 and 5.15 show two representative voxel-level selection cases for Voxels 177 and 164, respectively. Voxel 177 exhibits a richer candidate distribution, with several competitive trade-offs between stored pose quality and local dexterity. In contrast, Voxel 164 presents a sparser distribution, reflecting more restrictive local kinematic conditions and fewer competitive feasible entries. In both cases, the selected configuration corresponds to a best-compromise entry within the Pareto set, balancing residual accuracy and manipulability according to the database content available in that voxel. This comparison shows that the same selection rule can be applied to voxels with different candidate-set structures.

The selected trade-offs can also be examined through the internal PCC parameters of the best-compromise configurations. Figures 5.16 and 5.17 report the signed curvature, bending-plane angle, and twist profiles of the representative configurations selected for Voxels 177 and 164, respectively. The two profiles differ noticeably, indicating that different local workspace regions may require different internal TDCR shapes. This supports the role of the CAP framework not only as a reachability database, but also as a means of selecting dexterity-aware configurations among multiple feasible alternatives within a queried local neighbourhood.

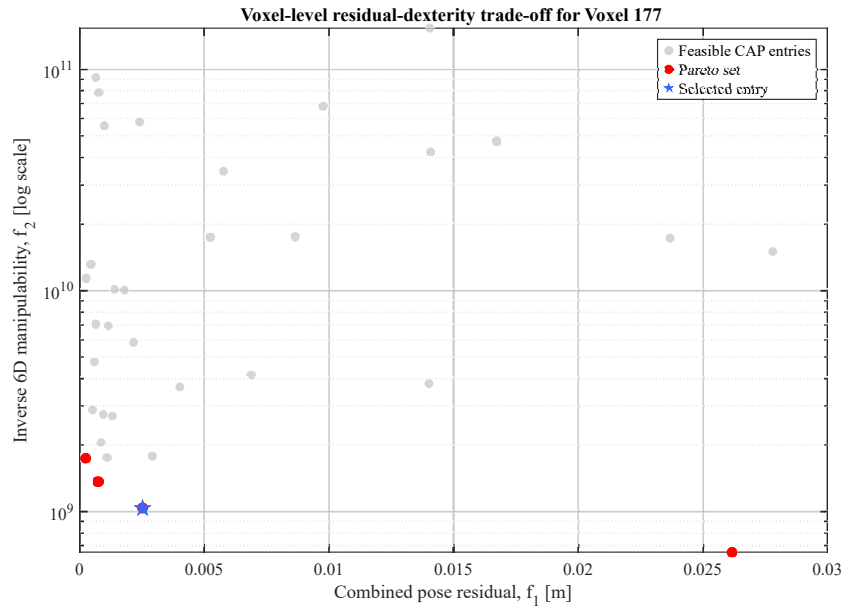


Figure 5.14 Voxel-level residual-dexterity trade-off for Voxel 177. The axes show the combined pose residual f_1 and inverse 6D manipulability f_2 . Gray points denote feasible CAP entries, red markers indicate the Pareto set, and the blue star marks the representative entry selected by the normalized ideal-point criterion.

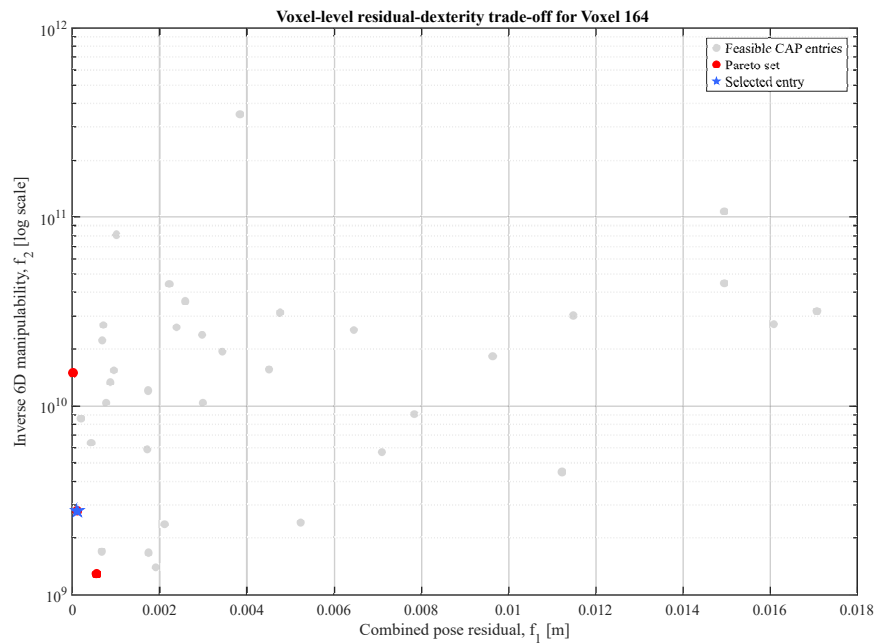


Figure 5.15 Voxel-level residual-dexterity trade-off for Voxel 164, showing a sparser feasible CAP distribution relative to Voxel 177. Axes, markers, and the normalized ideal-point selection criterion follow Figure 5.14.

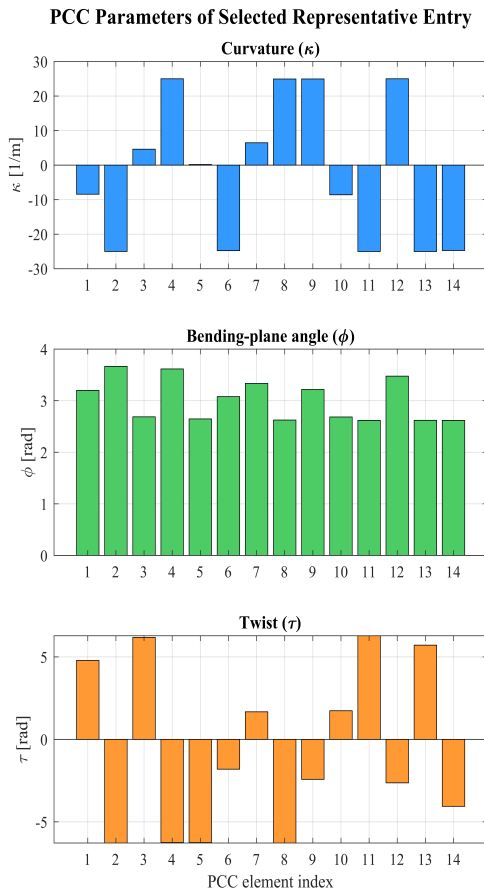


Figure 5.16 PCC parameters of the representative CAP entry selected for Voxel 177. The disk-wise curvature, bending-plane angle, and twist profiles define the internal TDCR configuration identified by the residual-dexterity Pareto/ideal-point criterion.

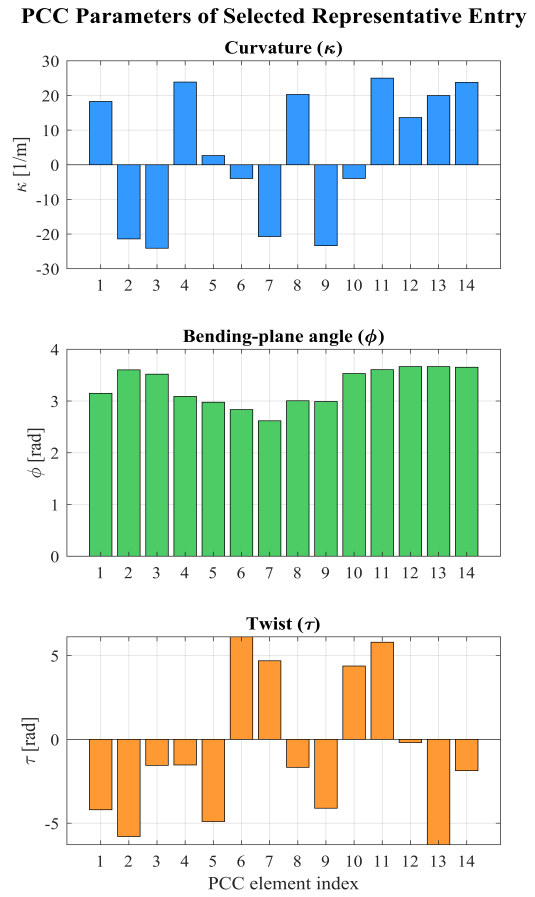


Figure 5.17 PCC parameters of the representative CAP entry selected for Voxel 164. The resulting internal TDCR configuration is distinct from that selected for Voxel 177. Parameter definitions and selection criterion follow Figure 5.16.

5.14 Experimental Validation

This section provides a proof-of-concept hardware evaluation of the CAP-based configuration-selection framework on the two-segment TDCR prototype. The objective is not to statistically validate the entire CAP database, but to assess the consistency between a selected database entry, the corresponding solver-found pose, and the experimentally measured end-effector pose in the evaluated case. The validation therefore examines whether a representative CAP-selected configuration can be executed under open-loop tendon actuation and whether the resulting pose remains consistent with the offline CAP prediction.

Experiments were carried out on the custom TDCR platform described earlier in this chapter. Motor commands were issued from MATLAB through the DYNAMIXEL communication interface, while end-effector poses were measured using the OptiTrack motion-capture system. The recorded data were processed through coordinate-frame alignment with the adopted z-down convention, quaternion normalisation, and time synchronisation, so that the experimental poses could be compared consistently with the target and solver-found CAP poses.

Hardware validation was performed on Voxel 151, selected as a representative reachable mid-workspace region with 184 accepted voxel–orientation entries in the CAP database. This candidate density provides a sufficiently rich local set for applying the residual–dexterity Pareto/ideal-point selector, while avoiding boundary voxels with sparse feasibility. For this voxel, the database-driven selector returned Sample 45141 as the best-compromise representative configuration according to the criterion introduced in Section 5.12. In line with the voxel-level formulation of the selector, this sample should be interpreted as a selected representative CAP entry within the queried local neighbourhood, rather than as the optimum for a user-specified exact pose. The corresponding tendon commands were applied to the prototype, and the resulting end-effector pose was recorded by the OptiTrack system. Figure 5.18 shows the experimental setup and the selected CAP entry used for validation.

Figure 5.19 compares the CAP target pose, the solver-found CAP pose, and the experimentally measured start and end poses in the common z-down frame. The start pose is included only as the reference configuration before actuation, while the main comparison concerns the final pose after execution. As shown in the figure, the measured end pose moves from the initial state toward the local neighbourhood of the CAP-predicted poses.

The corresponding position and orientation errors are reported in Figures 5.20 and 5.21. In Figure 5.20, the horizontal axis reports the CAP Sample ID, which is the database index of each matched voxel–orientation entry associated with Voxel 151. This identifier is used only

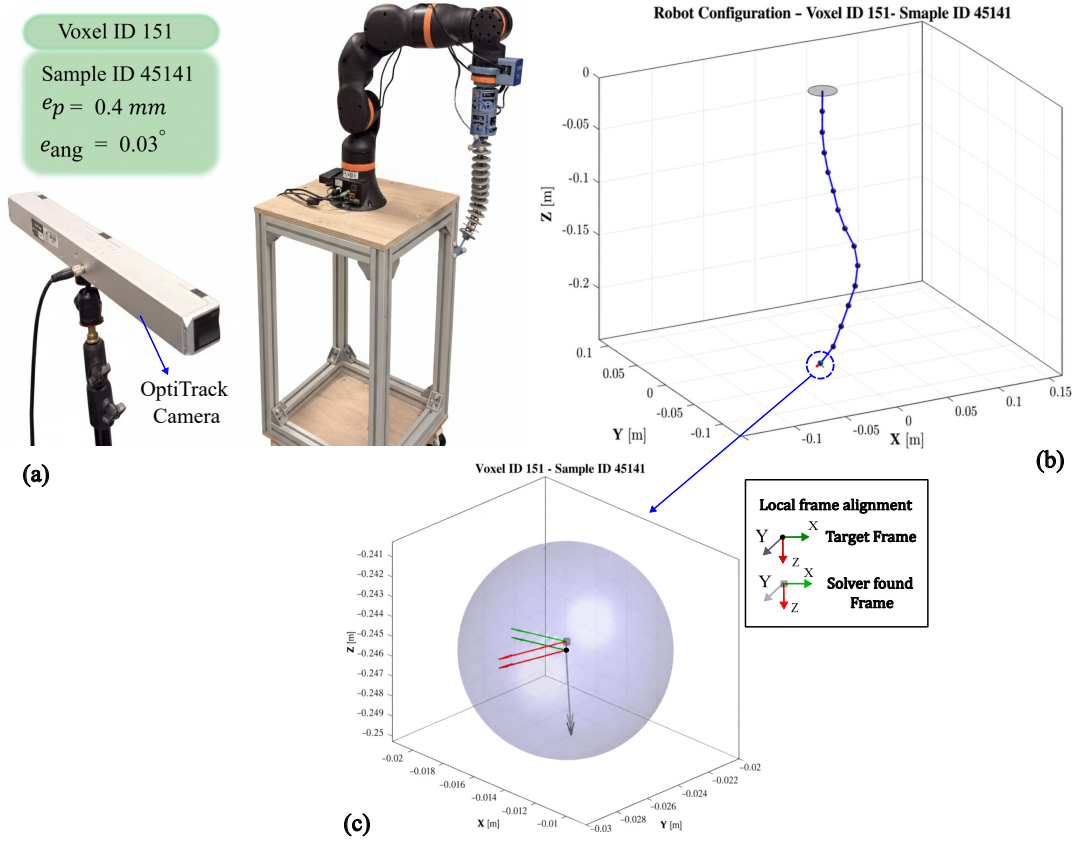


Figure 5.18 Hardware validation of the selected CAP entry. (a) TDCR–OptiTrack experimental setup. (b) Executed configuration for Voxel ID 151 and Sample ID 45141. (c) Target, solver-found, and experimentally measured poses within the selected voxel.

to trace the corresponding stored CAP entry and should not be interpreted as time, distance, orientation angle, or a spatial coordinate. Figure 5.20 therefore shows how the experimental errors vary across the matched CAP entries, while Figure 5.21 summarises the same errors in terms of their distributions.

The results show that position errors remain in the millimetre range, whereas orientation errors are generally larger but remain bounded. This behaviour is consistent with open-loop tendon actuation, where model simplifications and transmission effects may influence the achieved orientation more strongly than the end-effector position. An important observation is that the measured final pose is often closer to the solver-found CAP pose than to the nominal target pose. This can be explained by the discretised CAP formulation: the target pose corresponds to a voxel centre and an associated sampled orientation, while the stored CAP entry also contains the achieved pose returned by the offline IK solver for that specific query. Since the robot is actuated using the stored configuration, the experimental end pose is expected to be closer to the solver-found pose than to the idealised voxel–orientation target.

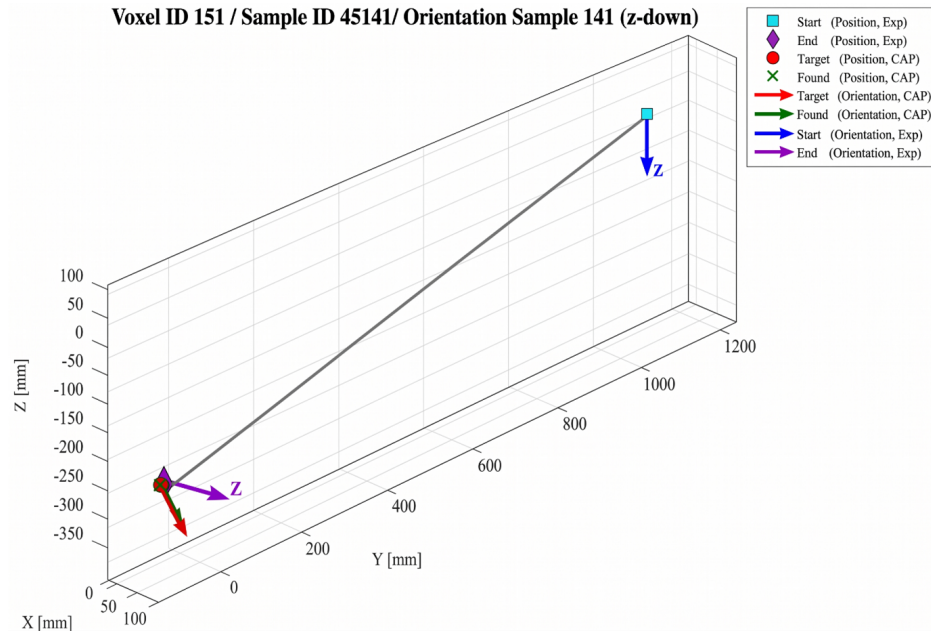


Figure 5.19 Spatial comparison of the CAP target, solver-found CAP pose, and experimentally measured start/end poses in Voxel 151.

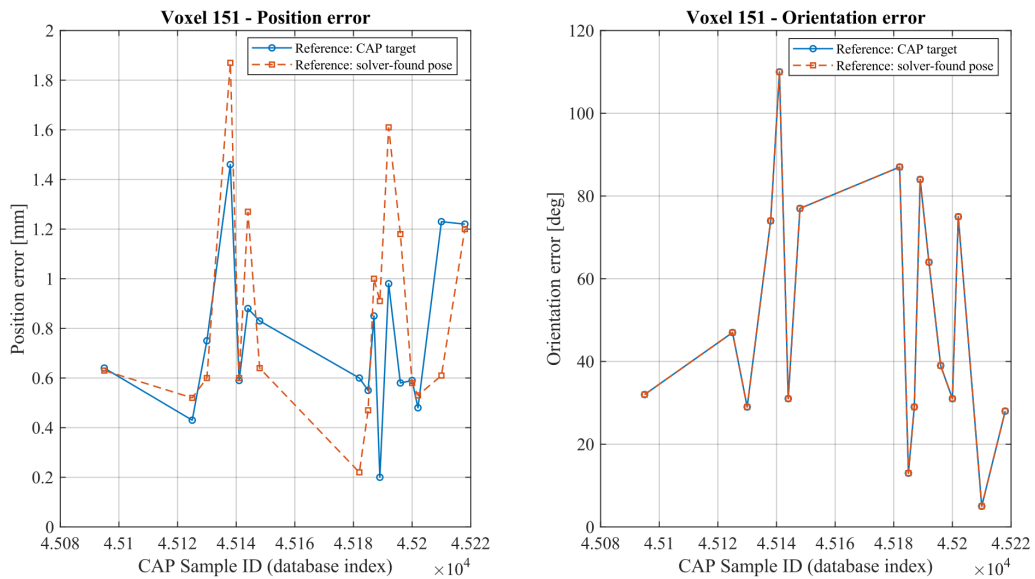


Figure 5.20 Position and orientation errors of the experimentally executed CAP samples associated with Voxel 151, reported with respect to both the nominal CAP target and the solver-found pose.

The same representative CAP-selected configuration was executed multiple times from the straight reference state to provide a qualitative indication of trial-to-trial consistency under open-loop tendon actuation. Although transmission effects introduce variability across trials, the measured end-effector poses remained in the neighbourhood of the stored solver-found configuration across the observed runs. These observations support the physical executability

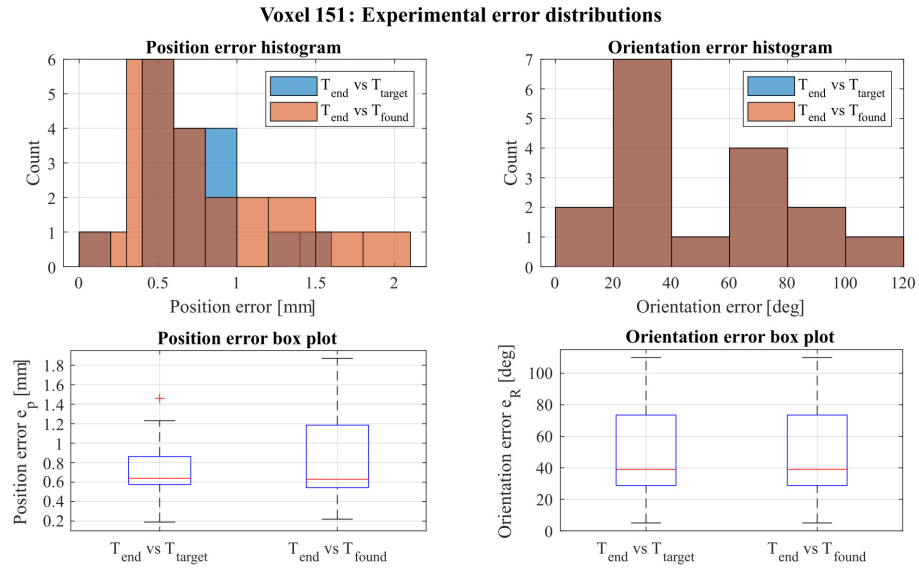


Figure 5.21 Experimental error distributions for Voxel 151. Position and orientation errors are summarized with respect to the nominal CAP target and the stored solver-found pose.

of the selected CAP entry in the evaluated voxel, but they should not be interpreted as a complete statistical repeatability characterisation of the full CAP database.

5.15 Chapter Summary

This chapter addressed the problem of generating accurate and executable task-space references for the two-segment tendon-driven continuum robot under the PCC parametrisation and actuation constraints established in Chapter 4. The chapter proceeded in three main steps, each addressing a specific limitation of the previous one.

First, a hybrid trajectory-generation framework was introduced, in which a supervised neural network provides fast initial estimates of the vertebra-level configuration and a constrained optimisation step refines the solution at each waypoint. The simulation results showed that this hybrid structure improves pose accuracy compared with purely learning-based inverse kinematics, while preserving coordinated tendon-length variations along the generated trajectories. However, this approach does not eliminate the need for repeated online optimisation. The computational effort still accumulates along the path, especially for longer trajectories, finer discretisations, or stricter pose requirements.

Second, this limitation motivated the transition to an offline capability-map formulation. The resulting CAP database associates discretised voxel–orientation targets in the distal workspace with feasible PCC configurations, residual pose errors, prototype-level actuation commands, and Jacobian-based dexterity descriptors. The database analysis showed that configurations associated with the same voxel are not equivalent in terms of local kinematic quality. Therefore, pose feasibility alone is not sufficient for runtime configuration selection. On this basis, a database-driven bi-objective selection strategy was introduced to rank feasible candidates within each queried voxel by combining pose accuracy and local dexterity through Pareto-set extraction and normalised ideal-point selection. The online stage is thereby reduced to database retrieval and ranking, without solving inverse kinematics or evaluating Jacobians online.

Third, a proof-of-concept hardware validation was performed on the physical TDCR prototype. The selected CAP entry for the evaluated voxel was executed under open-loop tendon actuation, and the measured end-effector pose was compared with both the nominal CAP target and the solver-found CAP pose. The results showed millimetre-range position errors in the evaluated case, while orientation errors were larger but remained bounded. The measured pose was often closer to the solver-found CAP pose than to the nominal voxel-level target, which is consistent with the discretised CAP formulation and the fact that the robot is actuated using the stored configuration.

At the same time, the framework has several limitations. The CAP database is constructed for a selected distal ROI and should not be interpreted as a complete description of the global

TDCR workspace. The experimental validation is limited to representative selected entries and does not constitute a full statistical validation of all database regions. Furthermore, the hardware execution remains open loop, so transmission effects, tendon compliance, friction, and calibration errors may influence the measured pose. These limitations define the direction for future improvements, including broader hardware evaluation, closed-loop execution, and extension of the database to richer task and contact conditions.

Overall, the chapter shows how the physically grounded model developed in Chapter 4 can be shifted from repeated online computation to an offline feasibility and dexterity database. This provides a practical bridge between model-based analysis and executable motion selection, and prepares the basis for the control-oriented developments considered in the next chapter.

Chapter 6

Prescribed-Time Fault-Tolerant Control of Tendon-Driven Continuum Robots

6.1 Motivation and Scope of the Chapter

Chapter 5 developed a planning layer that resolves kinematic redundancy and generates feasible, dexterity-aware reference trajectories using the capability map and hybrid inverse kinematics. This offline computation step ensures that only physically reachable and high-dexterity configurations are selected for execution. However, generating a feasible reference trajectory is only the first stage of the overall control problem. Once the reference is sent to the real robot, the execution layer must ensure accurate and safe tracking in the presence of the robot's inherent compliance, strong geometric nonlinearities, unmodeled elastic effects, external disturbances, actuator limitations, sensing imperfections, and possible time delays. For tendon-driven continuum robots, these challenges are significantly more demanding than in rigid-link manipulators because of the distributed deformation, configuration-dependent behavior, and strong coupling between bending coordinates [59, 60].

In recent years, continuum robots have attracted considerable attention in minimally invasive surgery and other confined-space applications. Their slender and compliant structure allows them to move through narrow anatomical pathways and interact more safely with delicate tissues, while providing a level of dexterity that conventional rigid instruments cannot easily achieve [59, 61–64]. At the same time, this compliance introduces major control difficulties. The robot dynamics are influenced not only by nonlinear geometry and elastic restoring effects, but also by internal friction, backlash, unmodeled disturbances, actuator saturation, measurement errors, and time-varying delays in the sensing and actuation

channels [61, 63, 65, 66]. As a result, a controller that performs well in simulation may not be sufficient to guarantee precise and safe behavior on the physical system.

A variety of control strategies have been studied for continuum and surgical robots. Early works relied on PID-type controllers because of their simple structure and ease of implementation, with later improvements through gain optimization and fuzzy tuning [66, 67]. Model predictive control has also been applied to tracking and deformation management, especially when state and input constraints must be handled explicitly [68–70]. These approaches remain useful in specific operating regimes, but their performance can deteriorate when the robot moves far from linearized conditions or when model uncertainty becomes significant. For highly nonlinear continuum robots, robust nonlinear and sliding-mode-based methods have therefore received increasing attention [65, 71]. Although these methods offer improved robustness, classical formulations still do not fully satisfy the transient and safety requirements of surgical applications.

In safety-critical tasks, it is not sufficient for the tracking error to converge asymptotically. The transient response itself must be regulated. The error should enter a safe region rapidly, within a known time, and without excessive overshoot or oscillation. This requirement explains the growing interest in finite-time, fixed-time, and prescribed-time control methods [65, 71–73]. Finite-time methods guarantee convergence in finite duration, but the settling time still depends on the initial condition. Fixed-time methods remove this dependence at the level of an upper bound. Prescribed-time methods go one step further by allowing the designer to specify the convergence time explicitly in advance, independently of the initial condition [65, 73]. For surgical robotics, where timing, precision, and safety must be addressed together, this property is particularly attractive.

At the same time, the controller must keep the tracking error within explicitly defined safety limits throughout the transient phase. Traditional prescribed-performance approaches can become restrictive when prescribed-time convergence, limited overshoot, and monotonic error evolution are required simultaneously [72, 74]. The monotone tube boundary approach addresses this limitation by converting the constrained tracking problem into an equivalent unconstrained stabilization problem while preserving monotonicity and designer-defined transient bounds [72]. This makes it especially suitable for continuum surgical robots, where even a temporary violation of a safe error region may lead to unsafe interaction with the environment.

To improve efficiency, this chapter also incorporates a learning-based adaptation mechanism. Its role is not to replace the model-based controller, but to refine its performance. The prescribed-time control law remains responsible for stability, transient regulation, and

keeping the error within safety bounds, while the learning component is used to compensate residual modelling errors online and reduce unnecessary control effort during tracking. This distinction is important in tendon-driven continuum robots, where excessive actuation is undesirable not only because of energy consumption, but also because it can increase tendon loading, heating, mechanical wear, and unsafe interaction forces in medical settings [73, 75–77].

Beyond nominal tracking, realistic operation also requires fault tolerance. In practice, sensor faults, actuator faults, time-varying delays, and saturation effects may occur simultaneously and interact with one another. A delayed measurement can affect the fault reconstruction process, saturation can prevent the intended compensating control action from being fully applied, and actuator faults can be partially masked by structural compliance or external load variation. For this reason, these non-idealities should not be treated as isolated effects. A practically useful controller must estimate and compensate for them within a unified framework. This observation motivates the second part of the chapter, where an observer-based prescribed-time fault-tolerant control strategy is developed to handle concurrent faults, delays, saturation, and unknown disturbances [77, 78].

The control study developed in this chapter addresses the same class of two-segment tendon-driven continuum robots considered throughout this thesis. For controller synthesis, a reduced-order dynamic model is adopted, which is consistent with standard practice in continuum-robot control and captures the dominant bending dynamics relevant to the present study. The proposed frameworks are validated numerically on a representative two-segment TDCR in order to assess their tracking, safety, and fault-tolerance properties under practical non-idealities. Experimental validation on the specific prototype developed in the previous chapters is left for future work.

This chapter therefore completes the execution layer of the thesis. Building directly on the planning results of Chapter 5, it develops two closely related control frameworks. The first is a safe and high-precision prescribed-time tracking controller based on monotone tube boundaries and enhanced by a learning-based optimal component. The second is its fault-tolerant extension, which actively addresses sensor and actuator faults, time-varying delays, saturation, and nonlinear uncertainty within a prescribed-time setting. In this way, the chapter moves from feasible motion generation to robust and safety-aware execution on the physical robot.

Section 6.3 introduces the prescribed-time monotone-tube tracking framework together with its learning-enhanced optimal component. Section 6.4 evaluates the performance of the nominal controller under prescribed-time safe-tracking conditions. Section 6.5 extends the

design to an observer-based prescribed-time fault-tolerant formulation that handles concurrent non-idealities. Section 6.6 reports the numerical evaluation of the fault-tolerant framework and its comparative performance. Finally, Section 6.7 summarizes the main findings of the chapter and clarifies how they support the overall thesis objective of achieving safe, precise, and practically deployable control for tendon-driven continuum robots [61, 65, 66, 72, 73, 78].

6.2 Mathematical Preliminaries and Dynamic Representation

The first part of this chapter addresses the nominal safe-tracking problem that forms the basis of the monotone-tube prescribed-time control framework. At this stage, the robot is considered under nominal operating conditions. Explicit actuator faults, sensor faults, time-varying delays, and saturation effects are intentionally excluded, since they belong to the fault-tolerant extension developed later in Section 6.5. The objective of the present section is therefore focused and precise: to define the robot coordinates used for control and to recall the control-oriented dynamic model adopted in the first study.

6.2.1 Bending Coordinates and Problem Setting

The control study developed in this chapter is based on the two-segment tendon-driven continuum robot introduced in the previous chapters, see Figure 6.1. Since the objective here is control design rather than geometric description, a reduced control-oriented model is adopted. Under the fixed-orientation assumption ($\delta_1 = \delta_2 = 0$), the dominant motion is represented by the two bending coordinates,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T \in \mathbb{R}^2. \quad (6.1)$$

Under this assumption, the generalized velocity is denoted by $\dot{\boldsymbol{\theta}}$, and the actuation input is represented by $\mathbf{u} \in \mathbb{R}^2$.

The desired reference trajectory is denoted by $\boldsymbol{\theta}_d(t)$, with first and second derivatives $\dot{\boldsymbol{\theta}}_d(t)$ and $\ddot{\boldsymbol{\theta}}_d(t)$, respectively. The tracking error and its derivative are defined as $\mathbf{e} = \boldsymbol{\theta} - \boldsymbol{\theta}_d$, $\dot{\mathbf{e}} = \dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_d$. For controller development, it is convenient to introduce the state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{\theta}^T & \dot{\boldsymbol{\theta}}^T \end{bmatrix}^T \in \mathbb{R}^4. \quad (6.2)$$

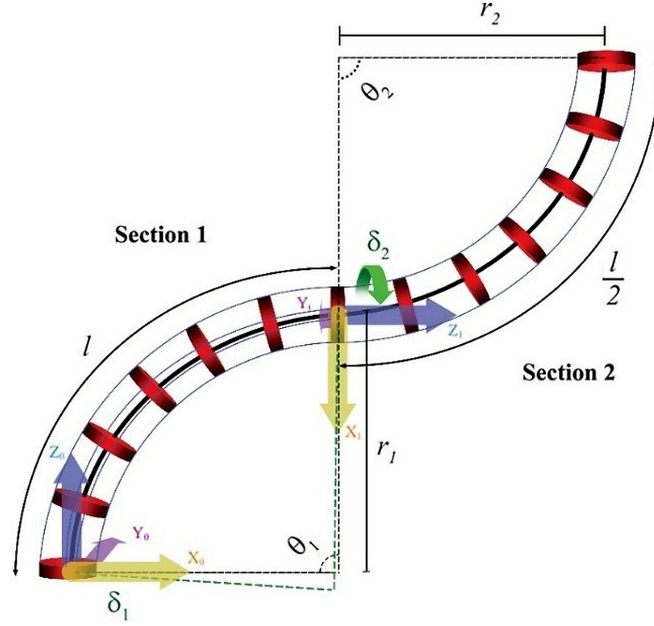


Figure 6.1 Two Segment Continuum Robot Structure.

6.2.2 Dynamic Model Used for Control Design

For control design, the robot dynamics in the bending coordinates are represented as

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\mathbf{v}(\dot{\theta}) + \mathbf{K}(\theta) = \mathbf{u}. \quad (6.3)$$

The matrices $\mathbf{M}(\theta)$, $\mathbf{C}(\theta, \dot{\theta})$, and $\mathbf{K}(\theta)$ correspond to the reduced-order Euler-Lagrange model of the same two-segment continuum robot considered in this thesis. Their explicit expressions can be found in [65]. Here, $\mathbf{M}(\theta) \in \mathbb{R}^{2 \times 2}$ is positive definite inertia matrix, $\mathbf{C}(\theta, \dot{\theta}) \in \mathbb{R}^{2 \times 3}$ collects velocity-dependent terms, $\mathbf{K}(\theta) \in \mathbb{R}^2$ denotes the elastic and gravity-related restoring effects, and $\mathbf{u} \in \mathbb{R}^2$ is the actuation input. The vector $\mathbf{v}(\dot{\theta})$ is introduced to compactly group the quadratic velocity terms appearing in the model. The vector of velocity monomials is defined as

$$\mathbf{v}(\dot{\theta}) = \begin{bmatrix} \dot{\theta}_1^2 & \dot{\theta}_1 \dot{\theta}_2 & \dot{\theta}_2^2 \end{bmatrix}^T. \quad (6.4)$$

This form is particularly convenient for controller synthesis because it makes the nonlinear coupling explicit while preserving a compact model structure. By defining the auxiliary variables $\mathbf{x}_1 = \theta$ and $\mathbf{x}_2 = \dot{\theta}$, the system can be written in state-space form as

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2, \quad \dot{\mathbf{x}}_2 = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{x}_1 = \theta. \quad (6.5)$$

The nominal $\mathbf{f}(\mathbf{x})$ and input gain $\mathbf{g}(\mathbf{x})$ are derived as

$$\mathbf{f}(\mathbf{x}) = -\mathbf{M}^{-1}(\theta) [\mathbf{C}(\theta, \dot{\theta})\mathbf{v}(\dot{\theta}) + \mathbf{K}(\theta)], \quad \mathbf{g}(\mathbf{x}) = \mathbf{M}^{-1}(\theta). \quad (6.6)$$

This state-space form is the one used throughout the controller development and numerical implementation presented in the first part of this chapter.

6.2.3 Control Objective

The objective of the nominal control framework developed in Section 6.3 is to design an input $\mathbf{u}$ such that the bending angles θ track $\theta_d(t)$ with three simultaneous properties. First, the transient tracking error must remain inside a predefined safe region determined by monotone tube boundaries. Second, convergence must occur within a prescribed settling time selected by the designer. Third, the control effort should be reduced as much as possible without compromising safety or prescribed-time convergence.

At this point, the problem is purely nominal. The model is assumed to be available in the form of Eq. (6.6), and no explicit delay channels, fault terms, or saturation blocks are introduced yet. This separation is intentional. The first part of the chapter establishes the safe prescribed-time tracking mechanism under nominal conditions. The second part then extends the same philosophy to a more realistic setting involving faults, delays, and additional non-ideal effects.

6.3 Prescribed-Time Safe Tracking via Monotone Tube Boundaries

This section develops the nominal safe-tracking controller used in the first part of the chapter. Based on the control-oriented model introduced in Section 6.2, the objective is to make the bending coordinates θ track the reference trajectory $\theta_d(t)$ while satisfying two requirements simultaneously. First, the tracking error must remain inside a designer-defined safe envelope throughout the transient phase. Second, the transformed closed-loop dynamics must enter a prescribed residual set within an assigned time T_p . To achieve this, the constrained tracking problem is first reformulated through monotone tube boundaries, then a prescribed-time sliding structure is introduced, and finally an actor-critic learning term is incorporated to improve control efficiency without altering the underlying safety mechanism.

6.3.1 Monotone Tube Boundary Formulation

Define the tracking error as

$$\mathbf{e}(t) = \theta(t) - \theta_d(t) = \begin{bmatrix} e_1(t) & e_2(t) \end{bmatrix}^T. \quad (6.7)$$

For each bending coordinate, the admissible error evolution is required to satisfy

$$b_{l,i}(t) < e_i(t) < b_{u,i}(t), \quad \forall i \in \{1, 2\}. \quad (6.8)$$

The tube boundaries are constructed from the monotone auxiliary function

$$q_i(t) = \begin{cases} \operatorname{csch}\left(q_{0i} + \frac{\alpha_i t}{T_{ci} - t}\right) + q_{\infty i}, & 0 \leq t < T_{ci}, \\ q_{\infty i}, & t \geq T_{ci}, \end{cases} \quad (6.9)$$

where $q_{0i} > 0$, and $\alpha_i > 0$, $T_{ci} > 0$ are design parameters. Here, $q_{\infty i}$ specifies the final admissible error radius, while T_{ci} determines the contraction time of the tube. The lower and upper boundaries are then defined as

$$b_{l,i}(t) = \operatorname{sign}(e_i(0)) (q_i(t) - q_{\infty i}) - \underline{\lambda}_i q_i(t), \quad (6.10)$$

$$b_{u,i}(t) = \operatorname{sign}(e_i(0)) (q_i(t) - q_{\infty i}) + \bar{\lambda}_i q_i(t), \quad (6.11)$$

with $\underline{\lambda}_i > 0$ and $\bar{\lambda}_i > 0$ controlling the transient width of the admissible tube. To reformulate the constrained tracking requirement, define the normalized error

$$\varepsilon_i(t) = \frac{e_i(t) - b_{l,i}(t)}{\Delta b_i(t)}, \quad \Delta b_i(t) = b_{u,i}(t) - b_{l,i}(t). \quad (6.12)$$

Then $0 < \varepsilon_i(t) < 1$ whenever $e_i(t)$ remains strictly inside the tube. The logarithmic transformation and the transformed error vector are introduced as

$$z_{e,i}(t) = \ln\left(\frac{\varepsilon_i(t)}{1 - \varepsilon_i(t)}\right), \quad i = 1, 2, \quad (6.13)$$

$$\mathbf{z}_e(t) = \begin{bmatrix} z_{e,1}(t) & z_{e,2}(t) \end{bmatrix}^T. \quad (6.14)$$

This transformation is the key step in the safety construction. Boundedness of $z_{e,i}(t)$ implies $0 < \varepsilon_i(t) < 1$, and therefore guarantees that the original tracking error does not leave the prescribed tube. In this way, transient safety in the original coordinates is converted into a boundedness problem in the transformed coordinates $\mathbf{z}_e$.

Differentiating Eq. (6.13) yields

$$\mathbf{R}_e(\mathbf{e}, t) = \operatorname{diag}(r_1, r_2), \quad r_i = \frac{1}{\Delta b_i \varepsilon_i (1 - \varepsilon_i)}. \quad (6.15)$$

$$\dot{\mathbf{z}}_e = \mathbf{R}_e(\mathbf{e}, t) \dot{\mathbf{e}} + \boldsymbol{\omega}(\mathbf{e}, t). \quad (6.16)$$

The vector $\omega(\mathbf{e}, t)$ collects the terms introduced by the time variation of the tube boundaries, with components

$$\omega_i = -r_i (\dot{b}_{l,i} + \varepsilon_i \Delta \dot{b}_i), \quad i = 1, 2. \quad (6.17)$$

Eq. (6.16) provides the dynamics of the transformed error vector $\mathbf{z}_e$, which will be used in the controller design.

6.3.2 Prescribed-Time Sliding Structure

To shape the transient of the transformed error, define the sliding variable

$$\mathbf{s} = \dot{\mathbf{z}}_e + \mu(t)\mathbf{z}_e, \quad (6.18)$$

where the time-varying gain is selected as

$$\mu(t) = \begin{cases} \frac{\sigma}{T_p - t}, & t < T_p, \\ \delta, & t \geq T_p, \end{cases} \quad (6.19)$$

with $\sigma > 0$, $\delta > 0$, and $T_p > 0$. The choice of $\mu(t) = \sigma/(T_p - t)$ is deliberate. As t approaches T_p , the gain $\mu(t)$ increases unboundedly, which intensifies the corrective action on the transformed dynamics as the prescribed time is approached. In this way, the designer-specified time T_p enters the closed-loop dynamics explicitly and provides the mechanism used later in the stability analysis to establish prescribed-time convergence. Differentiating Eq. (6.18) gives

$$\dot{\mathbf{s}} = \ddot{\mathbf{z}}_e + \dot{\mu}\mathbf{z}_e + \mu\dot{\mathbf{z}}_e. \quad (6.20)$$

Then, from the transformed error dynamics in Eq. (6.16), one obtains

$$\ddot{\mathbf{z}}_e = \dot{\mathbf{R}}_e \dot{\mathbf{e}} + \mathbf{R}_e \ddot{\mathbf{e}} + \dot{\omega}. \quad (6.21)$$

Using the nominal model of Section 6.2,

$$\ddot{\mathbf{e}} = \ddot{\boldsymbol{\theta}} - \ddot{\boldsymbol{\theta}}_d = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} - \ddot{\boldsymbol{\theta}}_d, \quad (6.22)$$

and therefore

$$\dot{\mathbf{s}} = \mathbf{R}_e \mathbf{g}(\mathbf{x})\mathbf{u} + \mathbf{R}_e (\mathbf{f}(\mathbf{x}) - \ddot{\boldsymbol{\theta}}_d) + \dot{\mathbf{R}}_e \dot{\mathbf{e}} + \dot{\omega} + \dot{\mu}\mathbf{z}_e + \mu\dot{\mathbf{z}}_e. \quad (6.23)$$

Eq. (6.23) is the control-oriented sliding dynamics used in the sequel. At this point, the original constrained tracking problem has been converted into the stabilization of the transformed variable $\mathbf{z}_e$, with explicit prescribed-time shaping through $\mu(t)$.

6.3.3 Learning-Enhanced Optimal Compensation

To improve control efficiency, an optimal component is introduced over the prescribed-time sliding dynamics. Consider the performance index

$$J(\mathbf{s}) = \int_t^{+\infty} (\mathbf{s}^T \mathbf{P}_s \mathbf{s} + \mathbf{u}^T \mathbf{R}_u \mathbf{u}) d\tau, \quad (6.24)$$

where $\mathbf{P}_s = \mathbf{P}_s^T \geq 0$ and $\mathbf{R}_u = \mathbf{R}_u^T > 0$. Following the development in [65], the formal HJB-based optimal control law can be expressed as

$$\mathbf{u}^* = -\frac{1}{2} \mathbf{R}_u^{-1} \mathbf{g}^T(\mathbf{x}) \mathbf{R}_e^T \frac{dJ^*(\mathbf{s})}{d\mathbf{s}}. \quad (6.25)$$

Define

$$\mathbf{G}_1 = (\mathbf{R}_e \mathbf{g}(\mathbf{x}))^{-1}, \quad \mathbf{G}_2 = \frac{1}{2} \mathbf{R}_u^{-1} \mathbf{g}^T(\mathbf{x}) \mathbf{R}_e^T. \quad (6.26)$$

Then the optimal control can be rewritten in the equivalent form

$$\mathbf{u}^* = -\mu \mathbf{G}_1 \mathbf{s} - \mathbf{G}_1 \mathbf{J}_0(\mathbf{s}), \quad (6.27)$$

where

$$\mathbf{J}_0(\mathbf{s}) = -\mu \mathbf{s} + \mathbf{G}_1^{-1} \mathbf{G}_2 \frac{dJ^*(\mathbf{s})}{d\mathbf{s}}. \quad (6.28)$$

Since the exact solution of the HJB equation is not available in closed form for the present nonlinear system, $\mathbf{J}_0(\mathbf{s})$ is approximated online by a radial basis function neural network:

$$\mathbf{J}_0(\mathbf{s}) = \mathbf{W}^{*T} \Phi(\mathbf{s}) + \varepsilon(\mathbf{s}), \quad (6.29)$$

where $\mathbf{W}^*$ denotes the ideal weight matrix, $\varepsilon(\mathbf{s})$ is the approximation error, and

$$\Phi(\mathbf{s}) = [\phi_1(\mathbf{s}) \quad \cdots \quad \phi_l(\mathbf{s})]^T, \quad \phi_j(\mathbf{s}) = \exp\left(-\frac{\|\mathbf{s} - \mathbf{m}_j\|^2}{\rho_j^2}\right). \quad (6.30)$$

The critic network estimates the value-function gradient through

$$\frac{dJ^*(\mathbf{s})}{d\mathbf{s}} = \mu \mathbf{G}_2^{-1} \mathbf{G}_1 \mathbf{s} + \mathbf{G}_2^{-1} \mathbf{G}_1 \widehat{\mathbf{W}}_c^T \Phi(\mathbf{s}), \quad (6.31)$$

while the actor generates the implementable optimal term

$$\widehat{\mathbf{u}}^* = -\mu \mathbf{G}_1 \mathbf{s} - \mathbf{G}_1 \widehat{\mathbf{W}}_a^T \Phi(\mathbf{s}). \quad (6.32)$$

Here $\Phi(\mathbf{s})$ is a Gaussian radial basis function vector. To preserve the prescribed-time safety structure, a compensation term is added:

$$\mathbf{u}_c = -(\mathbf{R}_e \mathbf{g}(\mathbf{x}))^{-1} \left[\left(\frac{\gamma_c}{2} + \frac{\gamma_a}{2} \right) \mathbf{s} + \mathbf{R}_e (\mathbf{f}(\mathbf{x}) - \ddot{\theta}_d) + \dot{\mathbf{R}}_e \dot{\mathbf{e}} + \dot{\omega} + \mu \mathbf{z}_e + \mu \dot{\mathbf{z}}_e \right], \quad (6.33)$$

where $\gamma_a > 0$ and $\gamma_c > 0$ are design constants. It should be noted that the time-derivative terms appearing in $\mathbf{u}_c$, namely $\dot{\mathbf{R}}_e$, $\dot{\omega}$, and $\dot{\mu}$, are analytically available. In particular, the tube-boundary functions are explicitly defined by the designer, and the prescribed-time gain $\mu(t)$ is also known in closed form. Therefore, the compensation input does not require numerical differentiation in implementation. The overall nominal controller is therefore

$$\mathbf{u} = \hat{\mathbf{u}}^* + \mathbf{u}_c. \quad (6.34)$$

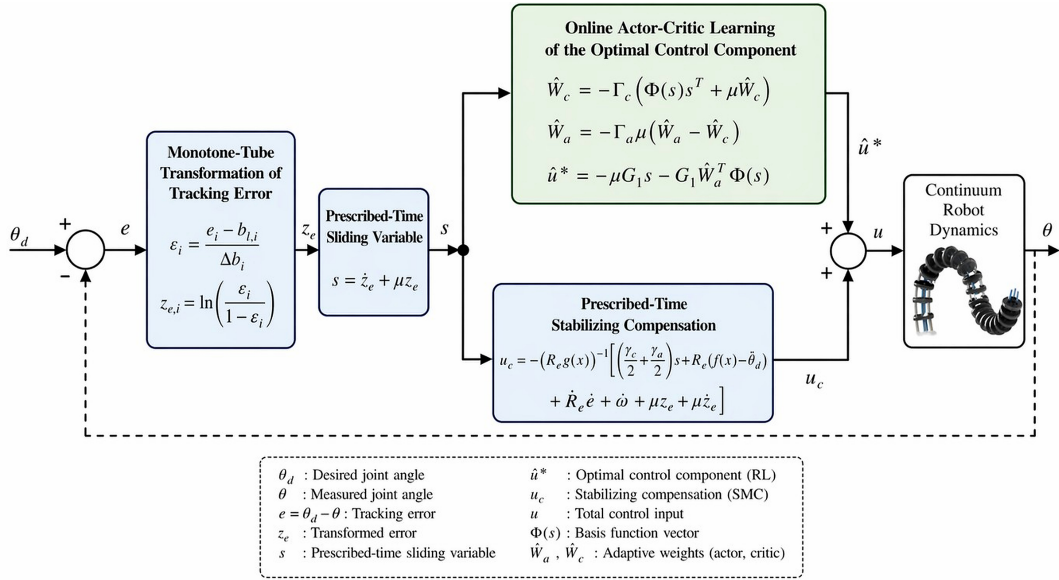


Figure 6.2 Block diagram of the proposed prescribed-time monotone-tube tracking controller.

Figure 6.2 summarizes the signal flow of the nominal controller. The reference bending trajectory θ_d is compared with the measured bending angles θ to generate the tracking error $\mathbf{e}$. This error is passed through the monotone tube transformation to obtain the transformed variable $\mathbf{z}_e$. The prescribed-time sliding surface $\mathbf{s} = \dot{\mathbf{z}}_e + \mu \mathbf{z}_e$ is then constructed. Based on this variable, two parallel branches are generated: the actor-critic reinforcement learning branch produces the optimal term $\hat{\mathbf{u}}^*$, while the compensation branch $\mathbf{u}_c$ guarantees prescribed-time safety and stability. The sum of these terms forms the total control input $\mathbf{u}$, which is applied to the continuum robot dynamics. The resulting bending angles θ are fed back to close the loop.

The actor and critic weights are updated according to

$$\hat{\mathbf{W}}_c = -\Gamma_c \left(\Phi(\mathbf{s}) \mathbf{s}^T + \mu \hat{\mathbf{W}}_c \right), \quad (6.35)$$

$$\hat{\mathbf{W}}_a = -\Gamma_a \mu \left(\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c \right), \quad (6.36)$$

where $\Gamma_c > 0$ and $\Gamma_a > 0$ are constant adaptation gain matrices.

In this structure, the compensation term $\mathbf{u}_c$ preserves safety and prescribed-time convergence, whereas $\hat{\mathbf{u}}^*$ improves control efficiency by reducing unnecessary actuation effort. This separation is important. The learning component refines the control action, but it does not carry responsibility for the core stability and safety mechanism. Figure 6.2 shows the overall block diagram of the proposed nominal controller.

6.3.4 Closed-Loop Stability Result

The main closed-loop stability property of the proposed nominal controller is summarized next. Before stating the result, the following standard prescribed-time residual-set lemma is recalled, since it is used directly in the proof.

Lemma 6.1 (PT-ISS-based residual-set convergence). Let $V(t)$ be a positive definite Lyapunov function satisfying

$$\dot{V} \leq -\kappa\mu(t)V + \mu(t)d \quad (6.37)$$

for some constants $\kappa > 0$ and $d > 0$, where $\mu(t)$ is the prescribed-time gain function adopted in this chapter. Then $V(t)$ remains bounded and enters a compact residual set no later than the prescribed time T_p , independently of the initial condition. More specifically, for all $t \geq T_p$, the Lyapunov function satisfies $V(t) \leq d/\kappa$. Consequently, if $V(t)$ contains the quadratic term $\frac{1}{2}\mathbf{s}^T\mathbf{s}$, then the sliding variable remains bounded after the prescribed time and satisfies $\|\mathbf{s}(t)\| \leq \sqrt{2d/\kappa}$. This conclusion follows from the standard prescribed-time input-to-state stability argument reported in [65].

Theorem 6.1. Consider the nominal control-oriented model of Section 6.2 together with the controller Eq. (6.34) and the adaptive laws Eqs. (6.35)–(6.36). Assume that $\theta_d(t)$, $\dot{\theta}_d(t)$ and $\ddot{\theta}_d(t)$ are bounded and sufficiently smooth, $\mathbf{g}(\mathbf{x})$ remains nonsingular in the operating region, the neural approximation error is bounded, and the initial tracking error satisfies the admissibility condition in Eq. (6.8). Then, for suitably selected monotone-tube parameters, prescribed-time parameters, and adaptive gains, all closed-loop signals remain bounded, the sliding variable $\mathbf{s}$ enters a compact residual set no later than T_p , and the tracking error satisfies Eq. (6.8).

Proof sketch. Consider the Lyapunov candidate

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} + \frac{1}{2}\text{trace}\left(\widetilde{\mathbf{W}}_c^T\Gamma_c^{-1}\widetilde{\mathbf{W}}_c\right) + \frac{1}{2}\text{trace}\left(\widetilde{\mathbf{W}}_a^T\Gamma_a^{-1}\widetilde{\mathbf{W}}_a\right), \quad (6.38)$$

where $\widetilde{\mathbf{W}}_a = \widehat{\mathbf{W}}_a - \mathbf{W}^*$, $\widetilde{\mathbf{W}}_c = \widehat{\mathbf{W}}_c - \mathbf{W}^*$. Using the closed-loop dynamics, the adaptive laws, and Young-type inequalities, the Lyapunov derivative can be upper bounded in the form

$$\dot{V} \leq -\kappa\mu(t)V + \mu(t)d \quad (6.39)$$

for some constants $\kappa > 0$ and $d > 0$. This inequality has the same structure as the prescribed-time input-to-state stability condition introduced earlier in the chapter. Therefore, by the corresponding lemma, $V(t)$ enters a compact residual set no later than T_p , independently of the initial condition. Since V bounds $\mathbf{s}$, $\widetilde{\mathbf{W}}_a$, and $\widetilde{\mathbf{W}}_c$, all these quantities remain bounded. Moreover, from $\mathbf{s} = \dot{\mathbf{z}}_e + \mu(t)\mathbf{z}_e$, one has $\dot{\mathbf{z}}_e = -\mu(t)\mathbf{z}_e + \mathbf{s}$. For $t < T_p$ and using $\mu(t) = \sigma/(T_p - t)$, its solution is given by $\mathbf{z}_e(t) = \left(\frac{T_p - t}{T_p}\right)^\sigma \mathbf{z}_e(0) + \int_0^t \left(\frac{T_p - t}{T_p - \xi}\right)^\sigma \mathbf{s}(\xi) d\xi$. Since $\mathbf{s}$ is bounded and $\sigma > 0$, it follows that $\mathbf{z}_e(t)$ remains bounded on $[0, T_p)$.

Since $\mathbf{s}$ is bounded and $\sigma > 0$, it follows that $\mathbf{z}_e(t)$ remains bounded on $[0, T_p)$. Consequently, each transformed error component $z_{e,i}$ remains finite, which implies $0 < \varepsilon_i < 1$ for all i . Therefore, the tracking error remains inside the prescribed monotone tube for all time. The detailed algebraic manipulations follow the controller derivation reported in [65].

Remark 6.1. The admissibility condition at $t = 0$ is not restrictive in practice. Since the monotone-tube boundaries are designer-defined through Eqs. (6.9)–(6.11), the initial tube width can be selected so as to contain the initial tracking error. In particular, the parameter q_{0i} can be chosen sufficiently large to ensure that $e_i(0)$ lies strictly inside the corresponding interval $(b_{l,i}(0), b_{u,i}(0))$ for each i . Therefore, the initial admissibility requirement can be enforced by design.

6.4 Numerical Evaluation of the Nominal Controller

This section evaluates the nominal prescribed-time safe-tracking controller developed in Section 6.3. The objective is to assess its transient safety, convergence behavior, and control effort under nominal operating conditions. The comparison is carried out against the benchmark controller reported in [79], which was reimplemented under the same reduced-order robot model and reference conditions for a fair evaluation.

6.4.1 Simulation Setting

The simulations are performed on the two-section continuum robot introduced in Section 6.2 using the same reduced bending model adopted in the controller design. In this part of

the chapter, the system is studied under nominal conditions, that is, without actuator faults, sensor faults, delays, or saturation effects.

In the first two cases, the prescribed settling time is set to $T_p = 0.25$ s. The monotone-tube parameters are chosen as $q_{01} = q_{02} = 2$, $q_{\infty 1} = q_{\infty 2} = 0.01$, $\alpha_1 = \alpha_2 = 1$, $T_{c1} = T_{c2} = 0.25$ s, and $\underline{\lambda}_1 = \underline{\lambda}_2 = \bar{\lambda}_1 = \bar{\lambda}_2 = 0.5$. The actor and critic adaptation gains are selected identically, and both neural networks use five radial basis functions with evenly distributed centres in $[-1, 1]$. These values are kept unchanged across the first two scenarios in order to show that the controller does not rely on repeated retuning when the reference profile changes. Three simulation cases are considered. The first examines tracking under a smooth sinusoidal reference. The second considers a step-ramp trajectory with abrupt changes. The third illustrates the effect of the prescribed-time and monotone-tube parameters on the transient response.

Case 1: Sinusoidal reference tracking.

In the first case, the desired bending trajectory is selected as

$$\theta_d(t) = \begin{bmatrix} 0.2 + \sin(2t) & 0.3 + \sin(2t) \end{bmatrix}^T. \quad (6.40)$$

The bending-angle responses are shown in Figure 6.3. The proposed controller drives both bending coordinates to the desired trajectory faster than the benchmark method, while maintaining a well-shaped transient response. The corresponding tracking errors are reported in Figure 6.4. The proposed method keeps the error within the prescribed monotone tube throughout the transient phase, whereas the benchmark controller exhibits a slower decay and temporary violations of the admissible region. In the reported simulations, the settling time is reduced from about 0.75 s for the benchmark method to about 0.2 s for the proposed controller.

The control inputs are shown in Figure 6.5. The proposed method achieves the same tracking task with smaller actuation effort, especially during the initial stage. The actor and critic weight evolutions, shown in Figure 6.6, remain bounded and settle smoothly, confirming stable online adaptation. Overall, this case shows that the proposed nominal controller provides faster convergence, safer transient behavior, and lower control effort than the benchmark method under smooth reference tracking.

Case 2: Step-ramp reference tracking.

The second case considers a step-ramp reference with abrupt changes. This scenario is more demanding because the controller must repeatedly recover tracking after sudden variations in the desired bending trajectory. The same controller parameters used in Case 1 are retained here. The bending-angle responses in Figure 6.7 show that the proposed controller

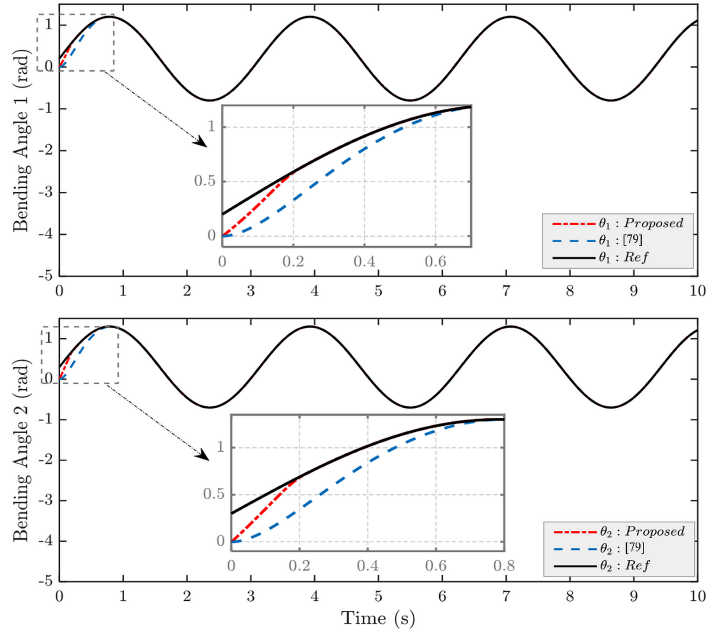


Figure 6.3 Comparison of continuum robot bending angles tracking under case 1.

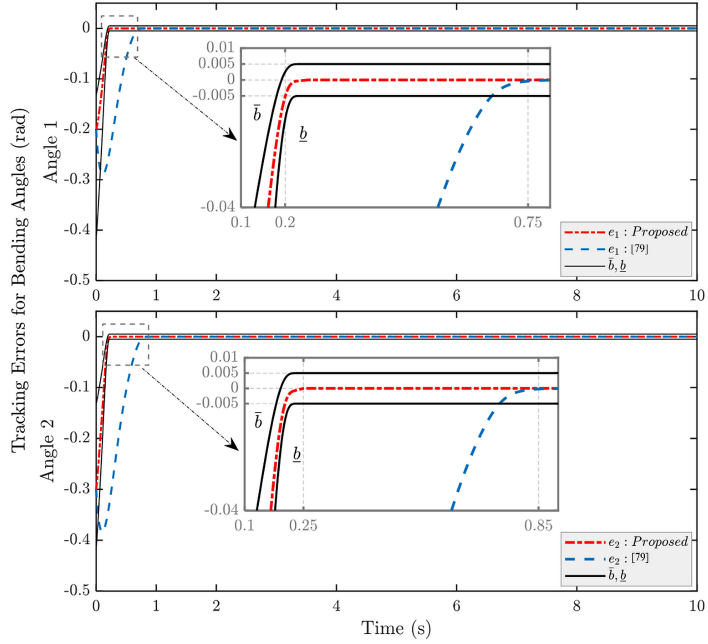


Figure 6.4 Comparison of continuum robot bending angles tracking error under case 1.

follows the successive reference changes more closely and recovers faster after each transition than the benchmark method. The safety advantage becomes clearer in the error plots of Figure 6.8. Even under abrupt trajectory variations, the proposed controller keeps the error inside the prescribed monotone tube, whereas the benchmark method exhibits larger excursions and temporary violations of the admissible region.

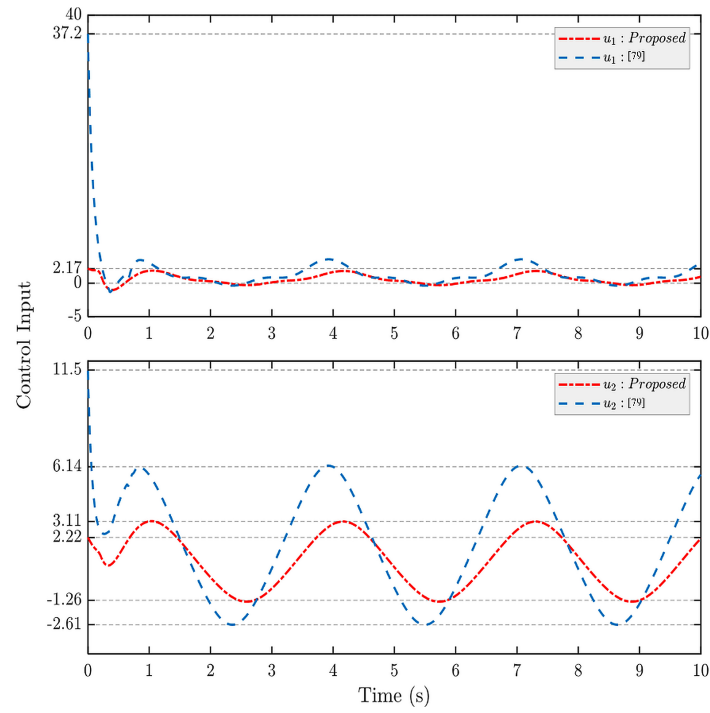


Figure 6.5 Comparison of control inputs under case 1.

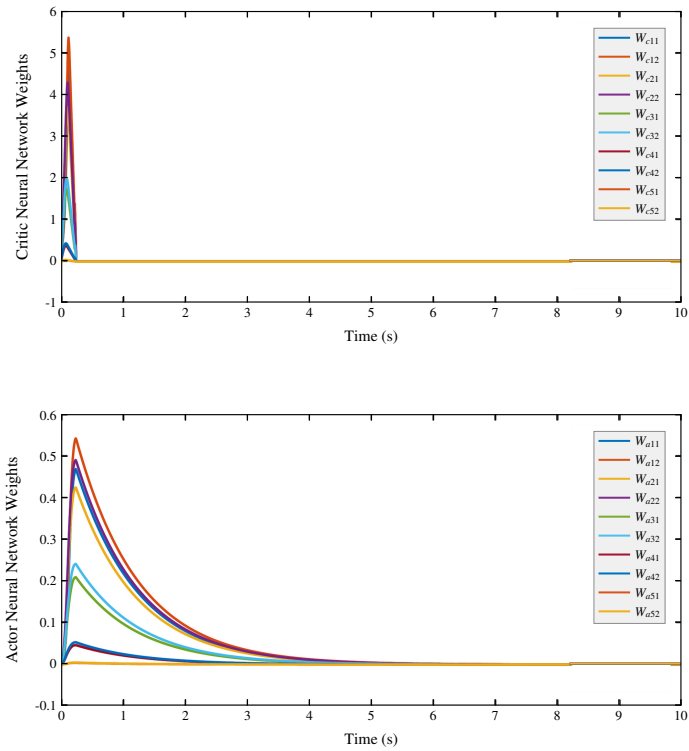


Figure 6.6 Evolution of Actor and Critic NN weights in case 1.

The corresponding control inputs are shown in Figure 6.9. Although this case requires stronger actuation than the sinusoidal one, the proposed method still uses smaller control

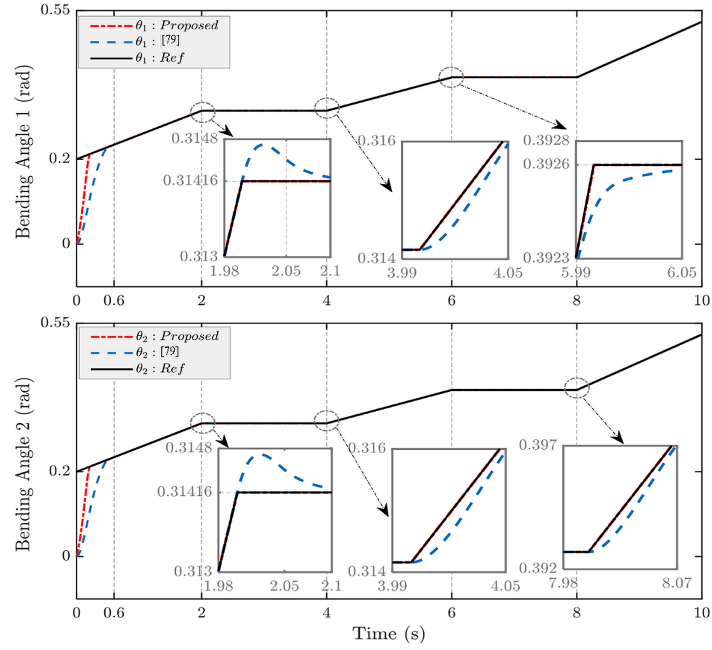


Figure 6.7 Comparison of continuum robot bending angles tracking under case 2.

magnitudes than the benchmark controller. This indicates that improved transient safety is achieved together with a more economical input profile.

These results show that the proposed nominal controller remains effective when the reference is no longer smooth and continues to combine safe transient behavior with reduced control effort.

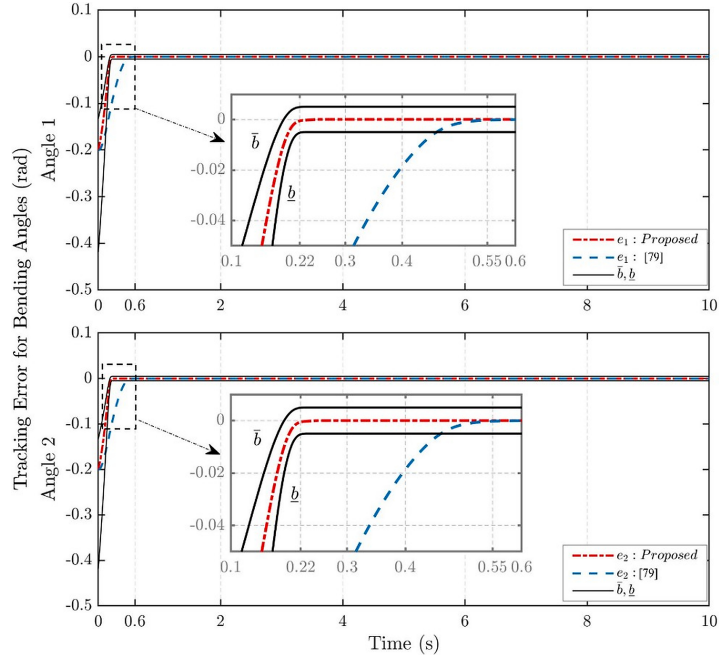


Figure 6.8 Comparison of continuum robot bending angles tracking under case 2.

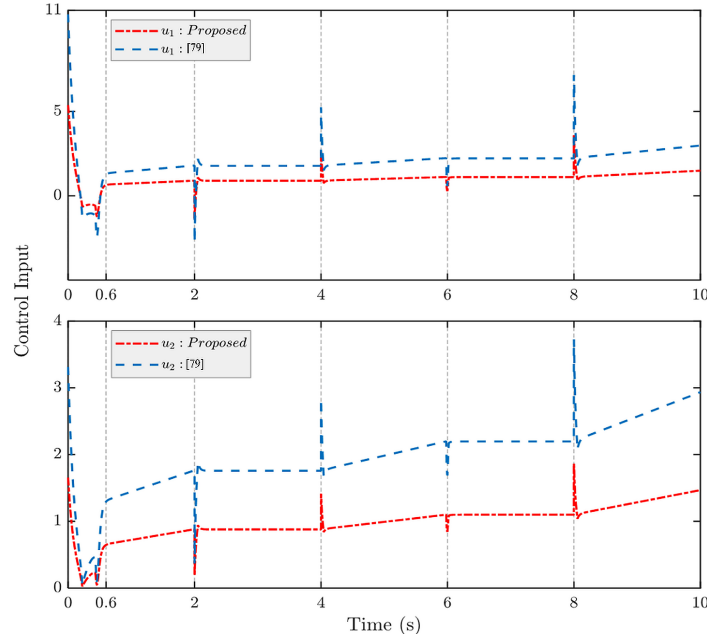


Figure 6.9 Comparison of control inputs under case 2.

Case 3: Effect of prescribed-time and tube parameters.

The third case illustrates the tuning flexibility of the proposed design. To modify the transient speed, the timing parameters are changed to $T_p = 0.5$ s, $T_{c1} = T_{c2} = 0.25$ s, while the remaining parameters are kept unchanged. The resulting responses are shown in Figures 6.10 and 6.11. As expected, the convergence becomes slower than in the previous cases. In the reported study, the effective settling behavior changes from about 0.2 s to about 0.4 s. At the same time, the steady-state tracking quality remains essentially unchanged, and the tracking error continues to evolve inside the admissible tube. This case shows that the prescribed-time and monotone-tube parameters can be used to reshape the transient response directly, without changing the overall controller structure. This is a useful property when different tasks require different convergence speeds.

Taken together, the three cases confirm the main features of the nominal controller: safe transient evolution inside the prescribed tube, fast and adjustable convergence, and lower control effort than the benchmark method.

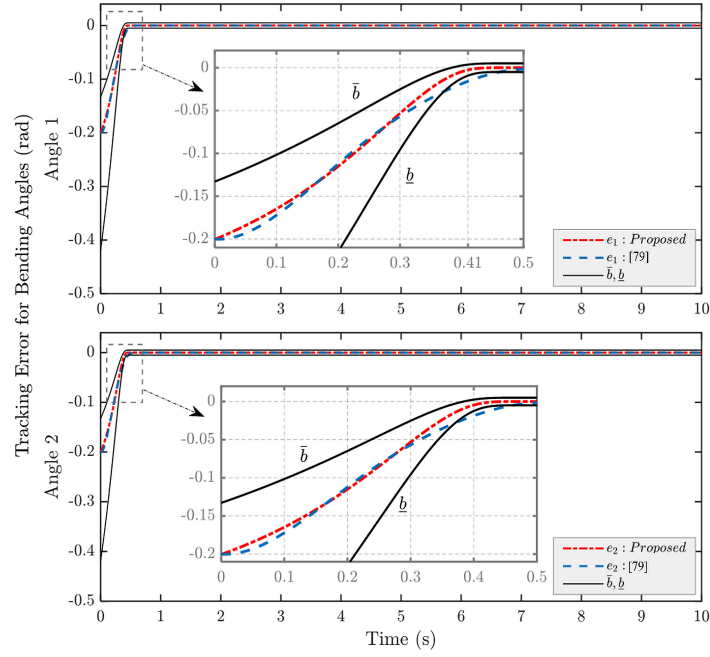


Figure 6.10 Comparison of continuum robot bending angles tracking error under case 3.

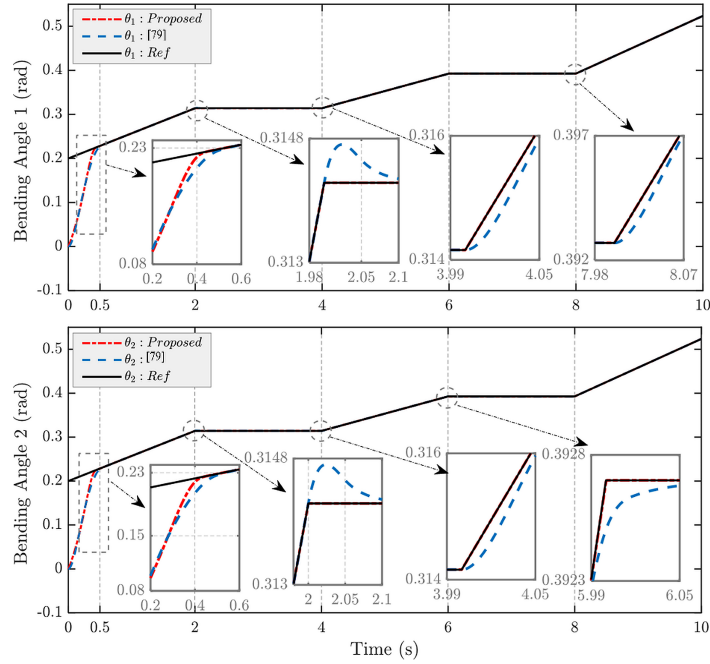


Figure 6.11 Comparison of continuum robot bending angles tracking under case 3.

6.5 Observer-Based Prescribed-Time Fault-Tolerant Formulation

The nominal tracking framework developed in Section 6.3 guarantees prescribed-time safe tracking under ideal operating conditions. In practice, however, tendon-driven continuum

robots are affected by actuator faults, sensor faults, time-varying input and output delays, actuator saturation, external disturbances, and unmodeled nonlinearities. Under such conditions, the measured output can no longer be treated as a reliable state-related signal, and the applied input may differ from the designed command. The nominal controller must therefore be extended to an observer-based fault-tolerant formulation.

This part of the chapter develops that extension. The proposed scheme combines a prescribed-time descriptor observer with a fault-tolerant control law that compensates for reconstructed faults, delays, saturation effects, and residual nonlinear uncertainty. Unlike the nominal monotone-tube formulation of Section 6.3, the present design guarantees prescribed-time convergence of an augmented tracking error to an adjustable compact residual set under non-ideal operating conditions.

6.5.1 Fault-Affected Model and Augmented Representation

Using the same control-oriented state vector $\mathbf{x} = [\theta^T \dot{\theta}^T]^T$ introduced in Section 6.2, the nominal reduced-order robot model is written in the compact form

$$\dot{\mathbf{x}} = \Lambda \mathbf{x} + \bar{\mathbf{G}}(\mathbf{x}) (\mathbf{u} + \varpi(\mathbf{x})), \quad \mathbf{y} = \mathbf{S} \mathbf{x}. \quad (6.41)$$

where $\Lambda = \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{I}_p \\ \mathbf{0}_{p \times p} & \mathbf{0}_{p \times p} \end{bmatrix}$, $\mathbf{S} = [\mathbf{I}_p \ \mathbf{0}_{p \times (n-p)}]$, $p = 2$, $n = 4$, and $\mathbf{y}$ denotes the healthy output. Here, $\bar{\mathbf{G}}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{p \times p} \\ \mathbf{g}(\mathbf{x}) \end{bmatrix}$ and $\mathbf{g}(\mathbf{x}) = \mathbf{M}^{-1}(\theta)$ represents the input distribution associated with the state-dependent gain introduced in Section 6.2, while $\varpi(\mathbf{x}) = -(\mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{K}(\theta))$ collects the nonlinear drift terms of the reduced robot dynamics. represents the input distribution associated with the state-dependent gain introduced in Section 6.2, while $\varpi(\mathbf{x}) = -(\mathbf{C}(\theta, \dot{\theta})\mathbf{v}(\dot{\theta}) + \mathbf{K}(\theta))$ collects the nonlinear drift terms of the reduced robot dynamics. To account for realistic operating conditions, the model is extended as

$$\begin{aligned} \dot{\mathbf{x}} &= \Lambda \mathbf{x} + \bar{\mathbf{G}}(\mathbf{x}) (\mathbf{u}_\tau + \varpi(\mathbf{x})) + \mathbf{G}_\omega \omega(t) + \mathbf{G}_{f_a} \mathbf{f}_a(t), \\ \mathbf{y}_o &= \mathbf{S} \mathbf{x} + \Theta^T \varphi(t), \end{aligned} \quad (6.42)$$

where $\mathbf{u}_\tau = \text{sat}(\mathbf{u}(t - \tau(t)))$ is the delayed and saturated input applied to the robot, $\tau(t)$ is the time-varying input delay, $\omega(t)$ denotes bounded external disturbances, and $\mathbf{f}_a(t)$ represents additive actuator faults. The matrices $\mathbf{G}_\omega$ and $\mathbf{G}_{f_a}$ are the known distribution matrices associated with the disturbance and actuator-fault channels, respectively. The measured output $\mathbf{y}_o$ is corrupted by the combined effect of output delay and sensor fault, namely

$$\Theta^T \varphi(t) = \mathbf{S} \zeta(t) + \mathbf{S}_{f_s} \mathbf{f}_s(t), \quad \zeta(t) = \mathbf{x}(t - h(t)) - \mathbf{x}(t), \quad (6.43)$$

where $h(t)$ is the time-varying output delay and $\mathbf{f}_s(t)$ is the additive sensor fault, and $\mathbf{S}_{f_s}$ is the sensor-fault distribution matrix. The term $\Theta^T \varphi(t)$ is therefore a compact representation of the overall output-channel corruption.

Since the output corruption term in Eq. (6.42) cannot be treated as a standard known input, the system is rewritten in an augmented form by introducing

$$\bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^T & \mathbf{x}_{f_s}^T \end{bmatrix}^T, \quad \mathbf{x}_{f_s} = \Theta^T \varphi(t). \quad (6.44)$$

This augmentation treats the output-channel delay and sensor-side corruption as additional quantities to be estimated online. In this way, the observer can reconstruct both the robot states and the output fault term within a unified prescribed-time framework, without requiring direct differentiation of corrupted measurements.

Since $\mathbf{x}_{f_s} = \Theta^T \varphi(t)$ represents an output-channel corruption term rather than a standard physical state with an explicit differential equation, the augmented model naturally takes a descriptor form. In this representation, the robot state $\mathbf{x}$ evolves dynamically, whereas $\mathbf{x}_{f_s}$ is treated as an algebraic quantity to be estimated online together with the state. This is the reason why the descriptor matrix $\mathbf{E}$ contains a singular block associated with the output-corruption component. The resulting formulation makes it possible to reconstruct both the robot state and the output-channel corruption within a unified observer framework without differentiating the corrupted measurement $\mathbf{y}_o$.

The control objective in this part of the chapter is therefore stronger than in the nominal case in terms of robustness, but different in its formal guarantee. The aim is to design an observer-based prescribed-time fault-tolerant controller such that all closed-loop signals remain bounded, the state and fault estimation errors converge within a designer-specified time, and the tracking error is driven to an adjustable compact residual set despite the simultaneous presence of actuator and sensor faults, input and output delays, saturation, disturbances, and model uncertainty.

6.5.2 Prescribed-Time Descriptor Observer

The fault-affected model introduced in Section 6.5.1 cannot be used directly for feedback, because the output corruption term $\mathbf{x}_{f_s}$ is unknown and the measured signal $\mathbf{y}_o$ cannot be treated as a reliable state-related quantity. A state-reconstruction mechanism is therefore required. The purpose of the observer developed in this subsection is to estimate both the robot state and the output-channel corruption term within a prescribed time, so that the subsequent control law can be formed using reconstructed rather than corrupted information. Using the augmented state introduced in Eq. (6.44), the fault-affected system is rewritten in

descriptor form as

$$\mathbf{E}\dot{\bar{\mathbf{x}}} = \bar{\Lambda}\bar{\mathbf{x}} + \bar{\mathbf{G}}_1(\mathbf{x})(\mathbf{u}_\tau + \bar{\boldsymbol{\omega}}(\mathbf{x})) + \bar{\mathbf{G}}_{f_a}\mathbf{f}_a(t) + \bar{\mathbf{G}}_\omega\omega(t) + \bar{\mathbf{N}}\mathbf{x}_{f_s}, \quad \mathbf{y}_o = \bar{\mathbf{S}}\bar{\mathbf{x}}, \quad (6.45)$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{I}_n & \mathbf{O}_{n \times p} \\ \mathbf{O}_{p \times n} & \mathbf{O}_{p \times p} \end{bmatrix}, \quad \bar{\Lambda} = \begin{bmatrix} \Lambda & \mathbf{O}_{n \times p} \\ \mathbf{O}_{p \times n} & -\mathbf{I}_p \end{bmatrix}, \quad \bar{\mathbf{S}} = \begin{bmatrix} \mathbf{S} & \mathbf{I}_p \end{bmatrix}, \quad \bar{\mathbf{N}} = \begin{bmatrix} \mathbf{O}_{n \times p} \\ \mathbf{I}_p \end{bmatrix}. \quad (6.46)$$

The term $\bar{\boldsymbol{\omega}}(\mathbf{x})$ denotes the state-dependent nonlinear drift inherited from the reduced robot model, whereas $\omega(t)$ represents the exogenous time-varying disturbance acting through the known distribution matrix $\mathbf{G}_\omega$. The augmented input and disturbance distribution matrices are defined as

$$\bar{\mathbf{G}}_1(\mathbf{x}) = \begin{bmatrix} \bar{\mathbf{G}}(\mathbf{x}) \\ \mathbf{O}_{p \times p} \end{bmatrix}, \quad \bar{\mathbf{G}}_\omega = \begin{bmatrix} \mathbf{G}_\omega \\ \mathbf{O}_{p \times m_1} \end{bmatrix}, \quad \bar{\mathbf{G}}_{f_a} = \begin{bmatrix} \mathbf{G}_{f_a} \\ \mathbf{O}_{p \times m_2} \end{bmatrix}. \quad (6.47)$$

The descriptor representation is introduced to incorporate the output-channel corruption term into the estimation problem without differentiating faulty measurements. In this way, the state and the sensor-side corruption can be reconstructed simultaneously within a single observer framework.

To obtain a form suitable for observer design, introduce the auxiliary variable

$$\boldsymbol{\pi} = \bar{\mathbf{x}} - \mathbf{H}^{-1} \mathcal{T} \mathbf{y}_o. \quad (6.48)$$

where $\mathcal{T} = \begin{bmatrix} \mathbf{O}_{n \times p} \\ \mathbf{Y} \end{bmatrix}$, $\mathbf{H} = \mathbf{E} + \mathcal{T}\bar{\mathbf{S}}$, $\mathbf{Y} = \text{diag}\{Y_1, Y_2, \dots, Y_p\}$, and $Y_i > 0$. Since the delayed input $\mathbf{u}_\tau$ is not directly available for reconstruction, the observer uses the delay-free saturated signal

$$\mathbf{u}_s = \text{sat}(\mathbf{u}(t)). \quad (6.49)$$

Following the approach in [65], the transformed composite uncertainty is approximated by an RBF neural network as

$$\bar{\boldsymbol{\omega}}(\bar{\mathbf{x}}, \mathbf{u}_\tau, \mathbf{u}_s) \approx \mathbf{W}^T \Phi(\bar{\mathbf{x}}, \mathbf{u}_s) + \bar{\boldsymbol{\varepsilon}}(\bar{\mathbf{x}}, \mathbf{u}_\tau, \mathbf{u}_s),$$

where $\mathbf{W}$ is the ideal weight matrix, $\Phi(\bar{\mathbf{x}}, \mathbf{u}_s)$ is the Gaussian basis vector, and $\bar{\boldsymbol{\varepsilon}}(\bar{\mathbf{x}}, \mathbf{u}_\tau, \mathbf{u}_s)$ is the bounded approximation error.

Unlike the scaling function used in Section 6.3 for prescribed-time tracking, the observer employs a prescribed-time time-base generator acting on the estimation dynamics. It is selected as

$$\rho_o(t) = \begin{cases} k_o \left(\frac{T_{p,o}}{T_{p,o} - t} \right)^{g_o}, & 0 \leq t < T_{p,o}, \\ \rho_{o,c}, & t \geq T_{p,o}, \end{cases} \quad g_o \geq 2, \quad (6.50)$$

where $T_{p,o}$ is the prescribed observer convergence time. The proposed prescribed-time descriptor observer is then given by

$$\begin{aligned} \dot{\hat{\pi}} = & \bar{\Lambda}_L \hat{\mathbf{x}} - \rho_o(t) \Pi \hat{\mathbf{x}} + \mathbf{H}^{-1} \hat{\mathbf{L}} \mathbf{y}_o + \bar{\mathbf{G}}_1(\mathbf{x}) \mathbf{u}_s + \bar{\mathbf{G}}_1(\mathbf{x}) \mathbf{v}_1 + \bar{\mathbf{G}}_\omega \mathbf{v}_2 + \bar{\mathbf{G}}_{f_a} \mathbf{v}_3 \\ & + \hat{\mathbf{W}}^T \Phi(\hat{\mathbf{x}}, \mathbf{u}_s) + \mathbf{H}^{-1} \mathbf{v}_\varepsilon + \bar{\mathbf{M}} \mathbf{v}_4, \end{aligned} \quad (6.51)$$

$$\hat{\mathbf{x}} = \hat{\pi} + \mathbf{H}^{-1} \mathcal{T} \mathbf{y}_o, \quad \bar{\Lambda}_L = \mathbf{H}^{-1} (\bar{\Lambda} - \mathbf{L} \bar{\mathbf{S}}), \quad \bar{\mathbf{M}} = \mathbf{H}^{-1} \bar{\mathbf{N}}. \quad (6.52)$$

Here, $\hat{\mathbf{x}}$ denotes the augmented state estimate, $\hat{\mathbf{L}}$ is the adaptive observer gain, $\hat{\mathbf{W}}$ is the estimated neural weight matrix, and $\Pi > 0$ is the prescribed-time design matrix.

The terms $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_\varepsilon$, and $\mathbf{v}_4$ are robust compensation signals introduced to cope with the unknown bounds associated with input-delay mismatch, external disturbances, actuator faults, approximation errors, and output-channel corruption, respectively. They are selected as

$$\begin{aligned} \mathbf{v}_1 = & -\hat{\rho}_u \text{sign}(\mathcal{Q}_1 \bar{\mathbf{S}} \tilde{\mathbf{x}}), \quad \mathbf{v}_3 = -\hat{\rho}_f \text{sign}(\mathcal{Q}_3 \bar{\mathbf{S}} \tilde{\mathbf{x}}), \\ \mathbf{v}_2 = & -\hat{\rho}_\omega \text{sign}(\mathcal{Q}_2 \bar{\mathbf{S}} \tilde{\mathbf{x}}), \quad \mathbf{v}_\varepsilon = -\hat{\rho}_\varepsilon \text{sign}(\mathcal{Q}_\varepsilon \bar{\mathbf{S}} \tilde{\mathbf{x}}), \\ \mathbf{v}_4 = & -\hat{\rho}_{f_s} \text{sign}(\mathcal{Q}_4 \bar{\mathbf{S}} \tilde{\mathbf{x}}). \end{aligned} \quad (6.53)$$

where the observer output error is $\tilde{\mathbf{y}} = \bar{\mathbf{S}} \hat{\mathbf{x}} - \mathbf{y}_o$. Since $\mathbf{y}_o = \bar{\mathbf{S}} \bar{\mathbf{x}}$, it follows that $\tilde{\mathbf{y}} = \bar{\mathbf{S}}(\hat{\mathbf{x}} - \bar{\mathbf{x}}) = \bar{\mathbf{S}} \tilde{\mathbf{x}}$. Therefore, the adaptive laws can be written directly in terms of the measurable innovation $\tilde{\mathbf{y}}$. The adaptive update laws for the unknown bounds are chosen as

$$\begin{aligned} \dot{\hat{\rho}}_u = & \gamma_u (\text{sign}(\mathcal{Q}_1 \tilde{\mathbf{y}})^T \mathcal{Q}_1 \tilde{\mathbf{y}} - \rho_o(t) \hat{\rho}_u), \\ \dot{\hat{\rho}}_\omega = & \gamma_\omega (\text{sign}(\mathcal{Q}_2 \tilde{\mathbf{y}})^T \mathcal{Q}_2 \tilde{\mathbf{y}} - \rho_o(t) \hat{\rho}_\omega), \\ \dot{\hat{\rho}}_f = & \gamma_f (\text{sign}(\mathcal{Q}_3 \tilde{\mathbf{y}})^T \mathcal{Q}_3 \tilde{\mathbf{y}} - \rho_o(t) \hat{\rho}_f), \\ \dot{\hat{\rho}}_\varepsilon = & \gamma_\varepsilon (\text{sign}(\mathcal{Q}_\varepsilon \tilde{\mathbf{y}})^T \mathcal{Q}_\varepsilon \tilde{\mathbf{y}} - \rho_o(t) \hat{\rho}_\varepsilon), \\ \dot{\hat{\rho}}_{f_s} = & \gamma_{f_s} (\text{sign}(\mathcal{Q}_4 \tilde{\mathbf{y}})^T \mathcal{Q}_4 \tilde{\mathbf{y}} - \rho_o(t) \hat{\rho}_{f_s}), \end{aligned} \quad (6.54)$$

while the observer gains and neural weights are updated according to

$$\begin{aligned} \dot{\hat{\mathbf{L}}} = & \Gamma_L (\mathcal{Q}_y \tilde{\mathbf{y}} \mathbf{y}_o^T - \rho_o(t) \hat{\mathbf{L}}), \\ \dot{\hat{\mathbf{W}}} = & \Gamma_w (\Phi(\hat{\mathbf{x}}, \mathbf{u}_s) \tilde{\mathbf{y}}^T \mathcal{Q}_w^T - \rho_o(t) \hat{\mathbf{W}}). \end{aligned} \quad (6.55)$$

Here, γ_u , γ_ω , γ_f , γ_ε , and γ_{f_s} are positive adaptation gains, whereas Γ_L and Γ_w are positive definite learning-rate matrices.

The observer structure can now be interpreted clearly. The descriptor transformation incorporates the output delay and sensor fault into the estimation model. The scaling term $\rho_o(t) \Pi \hat{\mathbf{x}}$ imposes prescribed-time decay on the estimation dynamics. The robust compensation signals $\mathbf{v}_1$ – $\mathbf{v}_4$ address the unknown bounded effects that cannot be measured

directly. Finally, the RBF approximation improves reconstruction of the uncertain nonlinear part of the transformed system. The prescribed-time convergence property of the observer is summarized as follows.

The observer design matrices are required to satisfy the following feasibility conditions:

$$\begin{aligned}
\mathbf{H}^{-\top} \mathcal{P} &= \mathcal{Q}_\varepsilon \bar{\mathbf{S}} = \mathcal{Q}_y \bar{\mathbf{S}}, \\
\bar{\mathbf{G}}_1^T(\mathbf{x}) \mathcal{P} &= \mathcal{Q}_1 \bar{\mathbf{S}}, \\
\bar{\mathbf{G}}_\omega^T \mathcal{P} &= \mathcal{Q}_2 \bar{\mathbf{S}}, \\
\bar{\mathbf{G}}_{fa}^T \mathcal{P} &= \mathcal{Q}_3 \bar{\mathbf{S}}, \\
\bar{\mathbf{M}}^T \mathcal{P} &= \mathcal{Q}_4 \bar{\mathbf{S}}, \\
\mathcal{P} &= \mathcal{Q}_\omega \bar{\mathbf{S}}, \\
\mathcal{P} \bar{\Lambda}_L + \bar{\Lambda}_L^T \mathcal{P} &= -\mathcal{Q} < 0.
\end{aligned} \tag{6.56}$$

It should be noted that $\bar{\mathbf{G}}_1^T(\mathbf{x})$ is state-dependent. In the observer feasibility conditions, this dependence is handled through a nominal constant representation adopted for the offline matrix design, while the residual state-dependent variation is absorbed into the bounded uncertainty and approximation terms of the observer model. These conditions are solved offline and guarantee the feasibility of the prescribed-time descriptor observer design.

Theorem 6.2. Consider the prescribed-time descriptor observer defined by Eqs. (6.51)–(6.55) for the fault-affected continuum-robot model in Eq. (6.45). Assume that the disturbance, actuator-fault, sensor-fault, and neural approximation-error terms are bounded, and that the observer design matrices satisfy the corresponding feasibility conditions stated above. Then all observer signals remain bounded, and the augmented estimation error

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} - \hat{\mathbf{x}} \tag{6.57}$$

converges to an adjustable compact residual set within the prescribed observer time $T_{p,o}$. Consequently, the state estimate $\hat{\mathbf{x}}$ and the output-channel corruption estimate $\hat{\mathbf{x}}_{f_s}$ become available in prescribed time for subsequent fault-tolerant control design.

Proof sketch. Define a composite Lyapunov function that includes the augmented estimation error, the adaptive bound-estimation errors, the observer-gain estimation error, and the neural-weight estimation error. By substituting the observer dynamics Eqs. (6.45)–(6.55) into the Lyapunov derivative, and by using the matrix equalities imposed by the observer design together with standard Young-type inequalities, the derivative can be upper bounded in the form

$$\dot{V}_o \leq -r_o \rho_o(t) V_o + \rho_o(t) d_o \tag{6.58}$$

for some constants $r_o > 0$ and $d_o > 0$. Then, by the prescribed-time input-to-state stability result introduced earlier in this chapter, the observer error enters a compact residual set no later than $T_{p,o}$. The detailed derivation follows the same Lyapunov steps reported in [80] and is omitted here for brevity. This observer provides the state and output-channel fault estimates required for the control design developed in the next subsection.

6.5.3 Composite Deep-Learning-Based Fault-Tolerant Control Law

The observer developed in Section 6.5.2 provides reconstructed state and output-corruption information in prescribed time. Based on these estimates, the next step is to design a fault-tolerant control law that preserves the prescribed-time philosophy of the chapter while explicitly handling actuator faults, sensor faults, input/output delays, saturation, disturbances, and residual nonlinear uncertainty. The control objective is therefore no longer limited to nominal safe tracking. Instead, the controller must maintain accurate trajectory tracking using reconstructed signals and must remain effective even when the applied input differs from the commanded one because of saturation and input-channel imperfections.

To improve steady-state tracking performance, an integral state is introduced. Define the augmented tracking state as

$$\chi = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\sigma} \end{bmatrix} \in \mathbb{R}^{n+p}, \quad \dot{\boldsymbol{\sigma}} = \mathbf{K}_I (\mathbf{S}\hat{\mathbf{x}} - \mathbf{r}). \quad (6.59)$$

where $\mathbf{r} \in \mathbb{R}^p$ is the reference bending trajectory, $\hat{\mathbf{x}}$ is the observer-based state estimate, and $\mathbf{K}_I \in \mathbb{R}^{p \times p}$ is a positive integral gain matrix. The integral state is initialized at $\boldsymbol{\sigma}(0) = \mathbf{0}$. It should be noted that $\boldsymbol{\sigma}$ is driven by the observer estimate $\hat{\mathbf{x}}$, rather than by the true state $\mathbf{x}$. Therefore, estimation errors affect the integral-state evolution during the initial transient. Nevertheless, since the observer converges in prescribed time and its error remains bounded, this coupling remains well behaved throughout the transient phase.

Using this definition, the augmented dynamics can be written in the compact form

$$\dot{\chi} = \mathcal{A}\chi + \mathcal{B}\mathbf{u} + \mathcal{B}\Delta\mathbf{u} - \mathcal{B}\mathbf{u}_d + \mathcal{B}_\omega\omega(t) + \mathcal{B}_{f_a}\mathbf{f}_a(t) + \mathcal{B}_{\mathcal{W}}\mathcal{W}(\mathbf{x}, \mathbf{u}_\tau) + \mathcal{B}_m\mathbf{r} - \check{\mathbf{K}}_I\mathbf{e}_x. \quad (6.60)$$

where $\mathbf{e}_x = \mathbf{x} - \hat{\mathbf{x}}$ is the state-estimation error, $\Delta\mathbf{u} = \mathbf{u}_s - \mathbf{u}$ denotes the mismatch introduced by input saturation, $\mathbf{u}_d = \mathbf{u}_s - \mathbf{u}_\tau$ represents the input-channel mismatch induced by delay, and $\mathcal{W}(\mathbf{x}, \mathbf{u}_\tau)$ collects the remaining uncertain nonlinear terms after observer-based

reconstruction. The augmented matrices are defined as

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \Lambda & \mathbf{0}_{n \times p} \\ \mathbf{K}_I \mathbf{S} & \mathbf{0}_{p \times p} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{0}_{p \times m} \end{bmatrix}, \quad \mathcal{B}_\omega = \begin{bmatrix} \mathbf{G}_\omega \\ \mathbf{0}_{p \times m_1} \end{bmatrix}, \quad \mathcal{B}_{f_a} = \begin{bmatrix} \mathbf{G}_{f_a} \\ \mathbf{0}_{p \times m_2} \end{bmatrix}, \\ \mathcal{B}_\mathcal{W} &= \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0}_{p \times n} \end{bmatrix}, \quad \mathcal{B}_m = \begin{bmatrix} \mathbf{0}_{n \times p} \\ -\mathbf{K}_I \end{bmatrix}, \quad \check{\mathbf{K}}_I = \begin{bmatrix} \mathbf{0}_{n \times n} \\ \mathbf{K}_I \mathbf{S} \end{bmatrix}. \end{aligned} \quad (6.61)$$

with the remaining matrices introduced in the same way from the corresponding terms of the fault-affected model. To define the tracking objective in dynamic form, a reference model is introduced as

$$\dot{\chi}_m = \mathcal{A}_m \chi_m + \mathcal{B}_m \mathbf{r}. \quad (6.62)$$

where $\mathcal{A}_m$ is chosen Hurwitz. As defined below, $\mathcal{A}_m = \mathcal{A} - \mathcal{B}\mathbf{K}$; therefore, this stability property is imposed through the appropriate selection of the baseline gain $\mathbf{K}$. This ensures that the reference model provides stable desired dynamics for the augmented tracking error.

The tracking error is then defined by

$$\mathbf{z} = \chi - \chi_m, \quad \hat{\mathbf{z}} = \hat{\chi} - \chi_m, \quad \hat{\chi} = \begin{bmatrix} \hat{\mathbf{x}} \\ \sigma \end{bmatrix}. \quad (6.63)$$

Following the same construction adopted in the source control design, $\mathcal{A}_m = \mathcal{A} - \mathcal{B}\mathbf{K}$, $\mathcal{A}_r = \mathcal{A} - \mathcal{B}\mathbf{K}_r$, where $\mathbf{K}$ and $\mathbf{K}_r$ are design gain matrices. Then, by adding the prescribed-time shaping term through the time-base generator $\rho(t)$, the tracking-error dynamics can be expressed in the form

$$\dot{\mathbf{z}} = \mathcal{A}_m \mathbf{z} + \rho(t) \mathcal{A}_r \mathbf{z} - \rho(t) \mathcal{A} \mathbf{z} + \mathcal{B}\mathbf{K} \hat{\chi} + \rho(t) \mathcal{B}\mathbf{K}_r \hat{\mathbf{z}} + \mathcal{B}\mathbf{u} + \Psi_c(t). \quad (6.64)$$

Here, $\Psi_c(t)$ denotes the lumped perturbation term that collects the saturation mismatch $\mathcal{B}\Delta\mathbf{u}$, the input-delay mismatch $-\mathcal{B}\mathbf{u}_d$, the disturbance and actuator-fault channels $\mathcal{B}_\omega \omega(t)$ and $\mathcal{B}_{f_a} \mathbf{f}_a(t)$, the residual nonlinear term $\mathcal{B}_\mathcal{W} \mathcal{W}(\mathbf{x}, \mathbf{u}_\tau)$, and the observer-error coupling term $\check{\mathbf{K}}_I \mathbf{e}_x$. These terms are treated as bounded non-ideal effects in the subsequent Lyapunov analysis. The prescribed-time scaling function used in the controller is selected in the same class as the observer scaling function, namely

$$\rho(t) = \begin{cases} k \left(\frac{T_p}{T_p - t} \right)^g, & 0 \leq t < T_p, \\ \rho_c, & t \geq T_p, \end{cases} \quad g \geq 2. \quad (6.65)$$

where T_p is the prescribed convergence time for the tracking loop. The role of $\rho(t)$ in the fault-tolerant design is analogous in spirit to that of $\mu(t)$ in the nominal controller,

although it acts through a different mechanism. As $t \rightarrow T_p^-$, $\rho(t)$ increases rapidly and amplifies the prescribed-time stabilizing terms appearing in the augmented tracking dynamics. Consequently, the corrective action becomes increasingly strong as the prescribed time is approached, which enables the controller to dominate the bounded non-ideal effects and to force the tracking error toward the prescribed compact residual set before T_p . For $t \geq T_p$, the scaling is frozen at the constant value ρ_c to avoid singular behavior. The control law is derived from an optimal-control viewpoint. Consider the quadratic performance index

$$J(\mathbf{z}) = \int_t^{+\infty} (\mathbf{z}^T \mathbf{P} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u}) d\tau, \quad (6.66)$$

where $\mathbf{P} = \mathbf{P}^T > 0$ and $\mathbf{R} = \mathbf{R}^T > 0$. The exact optimal solution of the associated HJB equation is not available in closed form because of the combined effects of faults, delays, saturation, and nonlinear uncertainty. For this reason, the implementable controller is written as a structured approximation consisting of a baseline stabilizing term, a prescribed-time correction term, an auxiliary saturation-compensation term, and a learned residual:

$$\mathbf{u}^* = -\mathbf{K}\hat{\chi} - \rho(t)\mathbf{K}_r\hat{\mathbf{z}} + \mathbf{K}_\Delta\Delta - \mathbf{J}_0(\mathbf{z}). \quad (6.67)$$

where $\mathbf{K}_\Delta \in \mathbb{R}^{p \times p}$ is the gain of the auxiliary compensator, $\Delta \in \mathbb{R}^p$ is its internal state, and $\mathbf{J}_0(\mathbf{z})$ denotes the unknown residual component of the optimal policy. To approximate this residual online, a deep neural network is introduced. The ideal residual component is a function of the true augmented tracking error $\mathbf{z}$. However, since the controller is implemented using observer-based estimates, the DNN is evaluated using the estimated tracking error. Therefore, define the augmented DNN input as $\hat{\mathbf{z}}_a = [\hat{\mathbf{z}}^T, 1]^T$, and write the implementable approximation in Eq. (6.68).

$$\mathbf{J}_0(\mathbf{z}) = \phi(\mathbf{z}_a) + \delta(\mathbf{z}), \quad (6.68)$$

where $\phi(\mathbf{z}_a)$ is the DNN output and $\delta(\mathbf{z})$ is the bounded approximation error. Accordingly, the implementable control law becomes

$$\mathbf{u} = -\mathbf{K}\hat{\chi} - \rho(t)\mathbf{K}_r\hat{\mathbf{z}} + \mathbf{K}_\Delta\Delta - \hat{\phi}(\hat{\mathbf{z}}_a). \quad (6.69)$$

The role of the DNN is not to replace the stabilizing controller. The stabilizing part is still carried by the feedback terms involving $\mathbf{K}$, $\mathbf{K}_r$, and $\mathbf{K}_\Delta$. The DNN only approximates the residual component that cannot be obtained analytically from the HJB equation. This distinction is important because it keeps the prescribed-time and robustness properties anchored in the model-based structure rather than in the learned approximation itself.

To compensate for saturation and residual input-channel mismatch, the auxiliary state Δ is updated according to the reconstructed healthy-output error. Before stating the update law,

define $\mathbf{e}_y = \mathbf{y}_o - \hat{\mathbf{x}}_{f_s} - \mathbf{S}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}_{f_s}$ denotes the estimated output-channel corruption term.

$$\dot{\Delta} = \Gamma_{\Delta} \left(-\sigma_{\Delta} \Delta - \rho(t) \Delta - \Delta \frac{\|\hat{\mathbf{z}}^T \mathbf{P} \mathcal{B} \Delta \mathbf{u}\| + \|\mathbf{e}_y^T \mathcal{M}^T \Delta \mathbf{u}\| + \alpha_6 \|\mathcal{M}^T \Delta \mathbf{u}\|^2}{\|\Delta\|^2} \right), \quad (6.70)$$

where $\Gamma_{\Delta} > 0$, $\sigma_{\Delta} > 0$, and $\alpha_6 > 0$ are design parameters.

The DNN weights are updated through a composite learning law that uses both the tracking error and the observer reconstruction error. For each network layer j , the update law is written as

$$\text{vec}(\dot{\hat{\vartheta}}_j) = \Gamma_{\vartheta_j} (-\sigma_{\vartheta_j} \text{vec}(\hat{\vartheta}_j) - \rho(t) \text{vec}(\hat{\vartheta}_j) + \Psi_j^T \mathcal{B}^T \mathbf{P} \hat{\mathbf{z}} + \Psi_j^T \mathcal{M} \mathbf{e}_y). \quad (6.71)$$

where $\hat{\vartheta}_j$ denotes the estimated weight matrix of layer j , $\Gamma_{\vartheta_j} > 0$ is the adaptation gain, $\sigma_{\vartheta_j} > 0$ is the leakage parameter, and Ψ_j is the corresponding regression matrix obtained from the DNN backpropagation structure. The first correction term in Eq. (6.71) is driven by the tracking error, whereas the second is driven by the observer output error. This is the reason the method is referred to as composite learning. The overall signal flow of the observer-based fault-tolerant architecture can be summarized by the block diagram of the proposed controller shown in Figure 6.12.

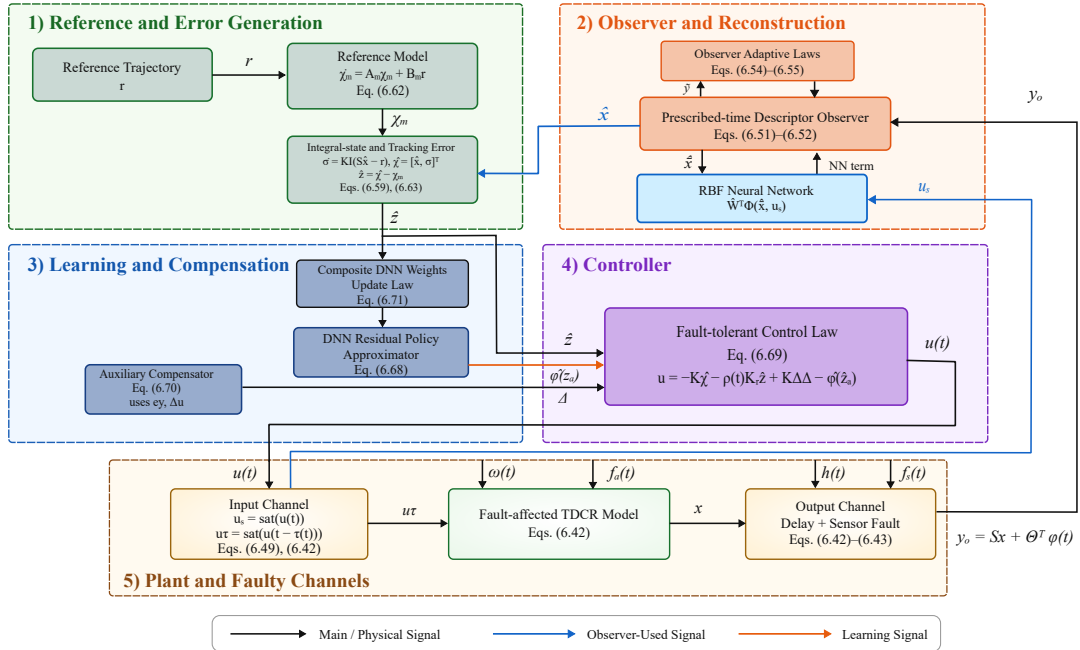


Figure 6.12 Block diagram of the proposed composite adaptive optimal control.

The reference trajectory $\mathbf{r}$ is first processed by the reference model to generate the desired augmented state χ_m , while the measured output $\mathbf{y}_o$, affected by output delay and sensor-side faults, is supplied to the prescribed-time descriptor observer. By combining the

adaptive compensation terms with the RBF neural approximation, the observer reconstructs the augmented state, providing in particular the state estimate $\hat{\mathbf{x}}$ and the output-channel corruption estimate $\hat{\mathbf{x}}_{f_s}$ within a prescribed time. The estimated state $\hat{\mathbf{x}}$ is then used by the integral-state generator to build the augmented state $\hat{\chi}$, and the tracking-error block computes $\hat{\mathbf{z}} = \hat{\chi} - \chi_m$. Based on $\hat{\mathbf{z}}$ and the observer reconstruction error, the composite DNN adaptation law updates the parameters of the residual-policy approximator, which generates an online estimate of the unknown residual term in the control law. In parallel, the auxiliary compensator produces the internal signal Δ to mitigate the effects of saturation and input-channel mismatch. These signals are combined in the composite deep-learning-based fault-tolerant control law to generate the command $\mathbf{u}$, which is then applied to the surgical continuum robot through the delayed and saturated input channel. The resulting measured output, after corruption by the output channel, is fed back to the observer, thereby closing the loop. In this way, the overall architecture integrates state reconstruction, uncertainty approximation, saturation compensation, and prescribed-time tracking within a unified control framework.

The prescribed-time stability property of the observer-based fault-tolerant control law is summarized as follows. The matrix $\mathcal{K}$ denotes the selection matrix associated with the physical-state component of the augmented state χ . The matrix $\mathcal{M}$ is introduced through the coupling condition $\mathcal{K}^T \mathbf{P} \mathcal{B} = \mathbf{S}^T \mathcal{M}^T$, and it appears later in the auxiliary compensator dynamics and in the composite DNN weight-update law. The controller design matrices are required to satisfy the following feasibility conditions:

$$\begin{aligned} \mathcal{K}^T \mathbf{P} \mathcal{B} &= \mathbf{S}^T \mathcal{M}^T & \sigma_{\Delta} - \beta_3 &> 0, \\ \mathbf{Q}_1 &= \mathcal{A}_m^T \mathbf{P} + \mathbf{P} \mathcal{A}_m + \alpha_s \mathbf{I} < 0, & \sigma_{\vartheta_j} - (\alpha_5 + \alpha_{14}) \mathbf{I} &> 0, \\ \mathbf{Q}_2 &= \mathcal{A}_r^T \mathbf{P} + \mathbf{P} \mathcal{A}_r - 2\mathbf{P} \mathcal{A} + \alpha_t \mathbf{I} < 0, & \mathfrak{g} = 1 - \alpha_{15} &> 0. \end{aligned} \quad (6.72)$$

Theorem 6.3. Consider the augmented tracking dynamics in Eqs. (6.60)–(6.65) associated with the fault-affected continuum-robot model Eqs. (6.42)–(6.44), together with the prescribed-time descriptor observer Eqs. (6.51)–(6.56), the fault-tolerant control law Eq. (6.69), the auxiliary compensator dynamics Eq. (6.70), and the composite DNN weight-update law Eq. (6.71). Assume that the actuator-fault, sensor-fault, disturbance, input-delay mismatch, saturation mismatch, observer error, and neural approximation error terms are bounded, and that the controller design matrices satisfy the feasibility conditions stated above. Then all closed-loop signals remain bounded, and the tracking error $\mathbf{z}$ converges to an adjustable compact residual set within the prescribed time T_p .

Proof sketch. Consider the Lyapunov candidate

$$V_c = \frac{1}{2} \mathbf{z}^T \mathbf{P} \mathbf{z} + \frac{1}{2} \Delta^T \Gamma_{\Delta}^{-1} \Delta + \frac{1}{2} \sum_{j=0}^K \text{vec}(\tilde{\vartheta}_j)^T \Gamma_{\vartheta_j}^{-1} \text{vec}(\tilde{\vartheta}_j). \quad (6.73)$$

where $\tilde{\vartheta}_j = \vartheta_j - \hat{\vartheta}_j$ denotes the DNN weight estimation error of the j -th layer. Using the tracking-error dynamics Eq. (6.64), the control law Eq. (6.69), the auxiliary compensator dynamics Eq. (6.70), the composite DNN update law Eq. (6.71), and the boundedness properties established by Theorem 6.2 for the observer, the derivative of V_c can be upper bounded in the form

$$\dot{V}_c \leq -r_c \rho(t) V_c + \rho(t) d_c \quad (6.74)$$

for some constants $r_c > 0$ and $d_c > 0$. By the same prescribed-time input-to-state stability argument used in the observer analysis, $V_c(t)$ enters a compact residual set no later than T_p . More specifically, for all $t \geq T_p$, the Lyapunov function remains bounded above by d_c/r_c , which implies that the augmented tracking error $\mathbf{z}$ remains bounded and, in particular, satisfies

$$\|\mathbf{z}(t)\| \leq \sqrt{\frac{2d_c}{r_c \lambda_{\min}(\mathbf{P})}}.$$

Consequently, $\mathbf{z}$, Δ , and the DNN weight errors remain bounded, and the tracking loop achieves prescribed-time fault-tolerant regulation. The full algebraic derivation follows the same sequence as reported in [80].

Remark 6.2. Unlike the nominal framework of Section 6.3, the fault-tolerant design developed in this section does not directly employ the logarithmic monotone-tube transformation. Instead, it guarantees prescribed-time convergence of the augmented tracking error to an adjustable compact residual set. Therefore, the fault-tolerant result should be interpreted as a bounded safety-oriented tracking guarantee under faults, delays, saturation, and uncertainty, rather than as a strict monotone-tube confinement result of the type established in the nominal case.

6.6 Numerical Evaluation of the Fault-Tolerant Controller

This section evaluates the observer-based prescribed-time fault-tolerant control framework developed in Section 6.5. The objective is to assess three aspects simultaneously: tracking accuracy under non-ideal operating conditions, reconstruction quality of the observer, and the control effort required to maintain stable motion. In contrast to the nominal study of

Section 6.4, the present simulations explicitly include input and output delays, actuator and sensor faults, actuator saturation, disturbances, and residual nonlinear uncertainty. The goal is therefore to verify that the proposed observer-controller architecture remains effective when the assumptions of the nominal case are no longer valid.

All simulations are carried out on the same two-segment reduced-order continuum-robot model introduced earlier in this chapter. The observer and controller parameters are selected according to the design procedure of Section 6.5. The performance is compared with representative robust benchmark methods. In particular, for the step and sinusoidal tracking tests, the comparison is made against the robust method in [81]. For the combined-fault resilience test, the comparison is made against the reinforcement-learning-based fault-tolerant strategy in [82]. In all cases, the same plant model, reference conditions, and fault-injection scenario are used to ensure a fair comparison.

6.6.1 Simulation Setting

The proposed framework consists of three coupled elements: the prescribed-time descriptor observer, the composite deep-learning-based fault-tolerant controller, and the auxiliary compensator that mitigates saturation and input-channel mismatch. The observer reconstructs the state and the output-channel corruption term from delayed and faulty measurements, while the controller uses these reconstructed quantities to maintain tracking. The DNN component does not replace the stabilizing feedback terms; instead, it approximates only the residual part of the control law that cannot be obtained analytically from the HJB formulation.

Two classes of reference trajectories are considered. The first is a multi-step bending reference used to evaluate transient recovery after abrupt changes. The second is a sinusoidal bending reference used to assess continuous tracking under simultaneous non-idealities. In the final test, simultaneous actuator faults, sensor faults, and time-varying delays are injected to examine fault resilience more directly. For the step test, the reference is selected as a three-stage profile:

$$r(t) = \begin{cases} \frac{\pi}{8}, & 0 \leq t < 10, \\ \frac{\pi}{2}, & 10 \leq t < 20, \\ \frac{\pi}{4}, & 20 \leq t \leq 30. \end{cases} \quad (6.75)$$

For the sinusoidal test, the desired bending trajectories are chosen as $r_1(t) = \frac{\pi}{4} + \frac{\pi}{6} \sin(t)$ and $r_2(t) = \frac{\pi}{4} + \frac{\pi}{6} \cos(t)$. These references are sufficiently different to reveal both transient and steady-state properties of the closed-loop system. The step profile stresses recovery from

abrupt variations, whereas the sinusoidal profile reveals the controller's ability to maintain smooth tracking under continuously varying conditions.

The main observer, controller, auxiliary-compensator, and neural-network parameters used in the fault-tolerant simulations are summarized in Table 6.1.

Table 6.1 Main parameters used in the observer-based fault-tolerant simulations.

Category	Parameters	Values
Observer	$\Pi, T_{p,o}, g_o, \rho_{o,c}, k_o, l$	$10^{-3}, 1\text{ s}, 3, 1, 10^{-10}, 10$
	Γ_L, Γ_ω	$10^{-3}, 10^{-3}$
	$\gamma_u, \gamma_\omega, \gamma_f, \gamma_\varepsilon, \gamma_{f_s}$	$10^{-3}, 10^{-6}, 10^{-3}, 10^{-3}, 10^{-3}$
Controller	$g, k, \mathbf{K}_I, \rho_c$	$3, 1, 1, 1$
	$\sigma_{\dot{\theta}_j}, \Gamma_{\dot{\theta}_j}$	$1, 1$
Auxiliary compensator	$\mathbf{K}_\Delta, \Gamma_\Delta, \sigma_\Delta, \alpha_6$	$1, 1, 1, 1$
Deep neural network	Hidden layers; neurons; activation functions; output layer	$3; (7, 8, 9); \tanh, \tanh, \text{logsig}; \text{linear}$

6.6.2 Step-Reference Tracking

The first test examines the response of the proposed fault-tolerant controller under the multi-step reference Eq. (6.75). Figures 6.13 and 6.14 report the bending-angle responses and the corresponding tracking errors, respectively. The proposed method drives both bending coordinates to the new reference values rapidly after each transition and exhibits a substantially faster decay of the tracking error than the benchmark method in [81]. In the reported simulations, the proposed framework achieves an effective settling time of approximately 0.3 s after each reference change, compared with approximately 5 s for the benchmark controller. This corresponds to an improvement of about 17 times and remains within the prescribed-time convergence bound specified by the control design.

This improvement is not limited to convergence speed. The error trajectories remain narrow and well-regulated during the switching instants, indicating that the observer-based controller preserves a controlled transient despite the simultaneous presence of delay, saturation, disturbance, and fault terms. By contrast, the benchmark method shows visibly larger excursions after each step and requires a longer recovery interval before the error becomes small again.

The corresponding control inputs are shown in Figure 6.15. Although abrupt reference changes naturally require stronger actuation, the proposed method still produces smaller and less oscillatory control signals than the benchmark controller. This is important in tendon-driven systems, where unnecessarily large control amplitudes may lead to higher tendon loading, increased wear, and reduced actuation efficiency. Figure 6.16 compares the

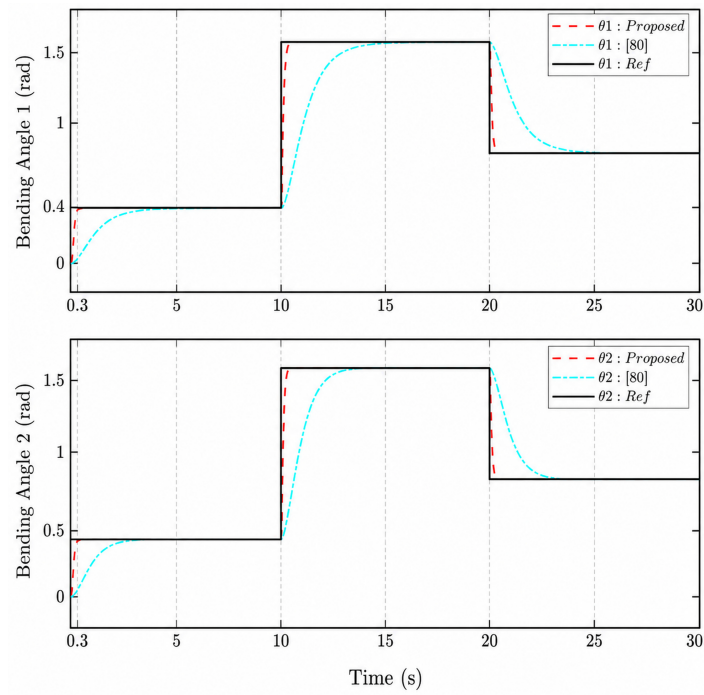


Figure 6.13 Comparison of continuum robot bending angles tracking under step signal.

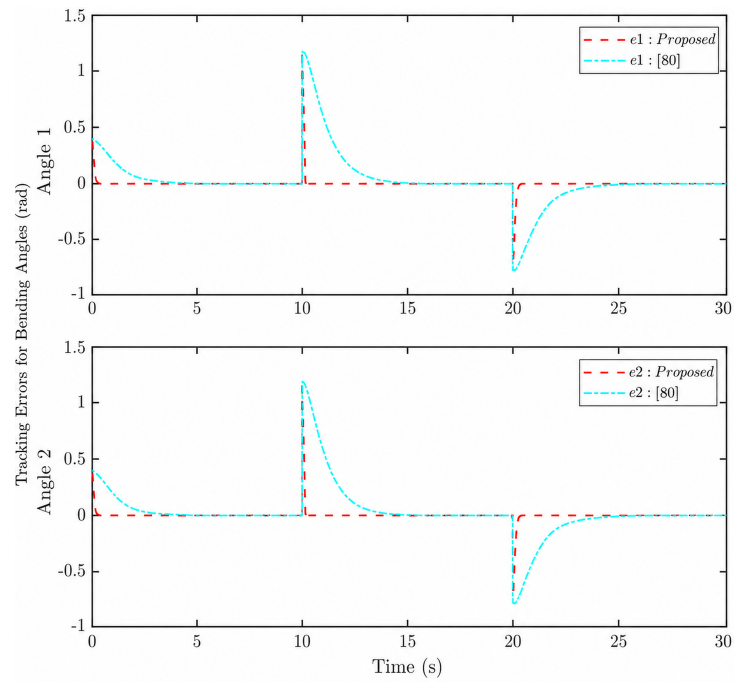


Figure 6.14 Comparison of continuum robot bending angles tracking error under step signal.

true robot states with their observer-based estimates. The close agreement confirms that the proposed observer reconstructs the state variables rapidly and accurately, even when the

measured output is corrupted. This reconstruction quality is a key reason why the controller can maintain stable tracking under adverse conditions.

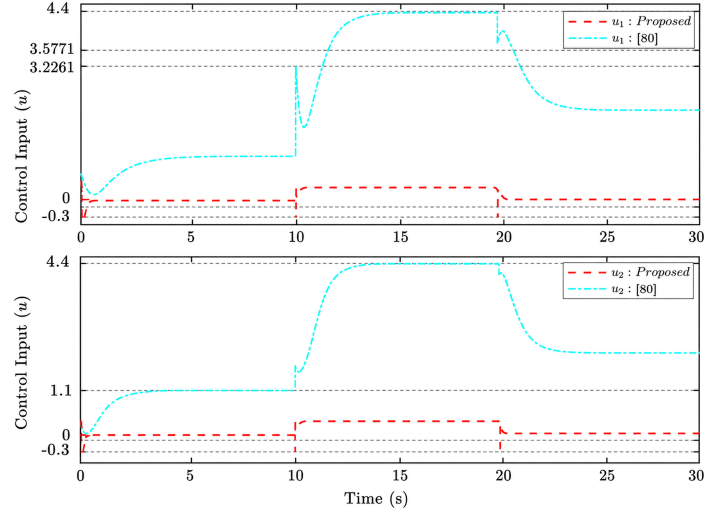


Figure 6.15 Comparison of control inputs under step signal.

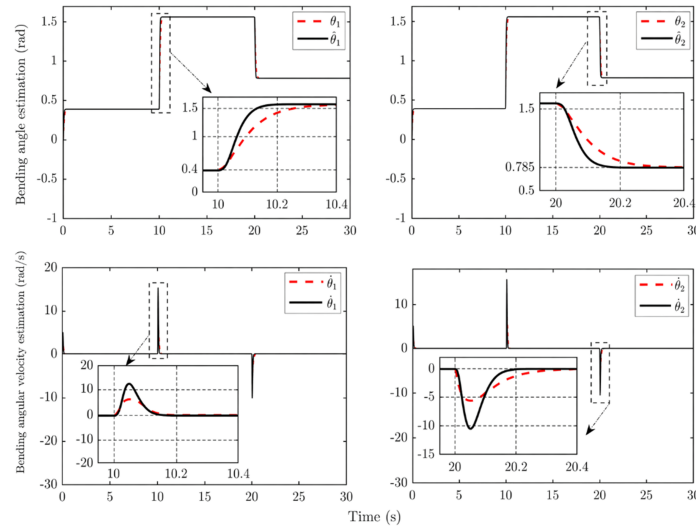


Figure 6.16 Dynamic variables of the continuum robot system and their estimated values under step signal.

Overall, the step-response test shows that the proposed prescribed-time fault-tolerant architecture combines fast recovery, accurate tracking, and moderate control effort. These three properties are especially important for surgical continuum robots, where abrupt motion corrections may occur during path updates or environment interaction.

6.6.3 Sinusoidal Tracking under Delays and Faults

The second test evaluates the proposed method under the sinusoidal references $r_1(t) = \frac{\pi}{4} + \frac{\pi}{6} \sin(t)$ and $r_2(t) = \frac{\pi}{4} + \frac{\pi}{6} \cos(t)$ introduced in Section 6.6.1. This case is more representative of practical continuum-robot motion, since the desired bending evolves continuously and the controller must compensate ongoing nonlinear effects rather than only isolated transients. Figures 6.17 and 6.18 show the bending-angle responses and the associated tracking errors.

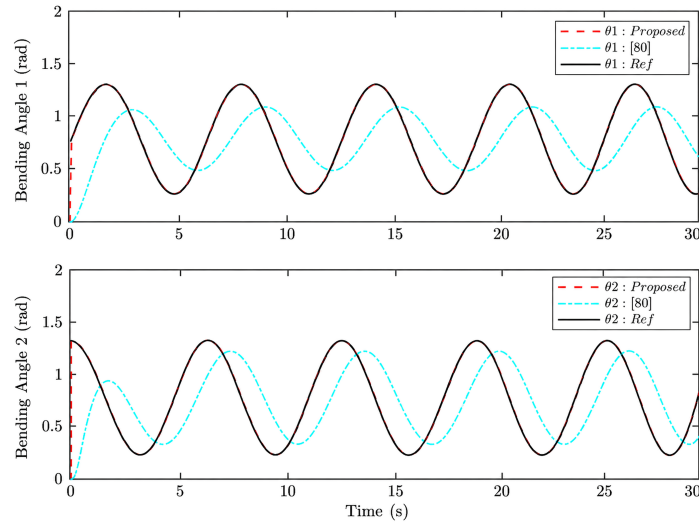


Figure 6.17 Comparison of continuum robot bending angles tracking under sinusoidal reference signal.

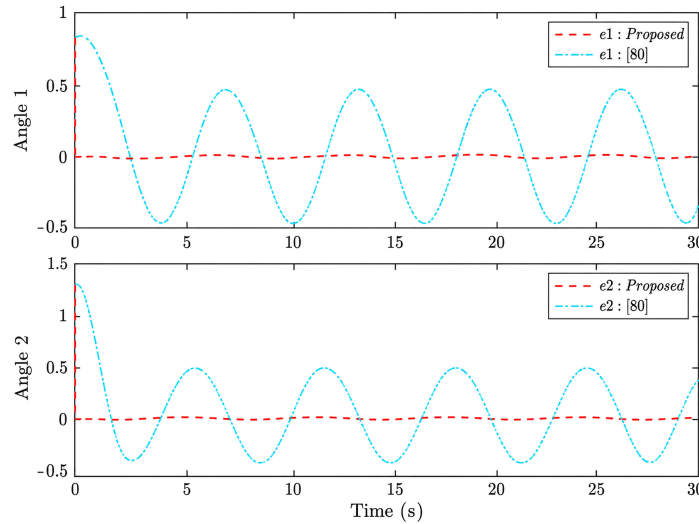


Figure 6.18 Comparison of continuum robot bending angles tracking error under sinusoidal signal.

The proposed method tracks both sinusoidal references closely throughout the simulation horizon, despite the simultaneous presence of input/output delays, disturbances, sensor corruption, and actuator-side imperfections. The resulting error envelopes remain small and

well behaved. In contrast, the benchmark method in [81] exhibits visibly larger deviations and more pronounced oscillatory behavior. This difference indicates that the proposed observer-based framework is not only effective for abrupt transient recovery, but also robust during sustained time-varying tracking.

The control inputs shown in Figure 6.19 further support this conclusion. The proposed controller generates smoother actuation profiles with lower amplitude and reduced oscillatory content than the benchmark strategy. From a practical viewpoint, this means lower mechanical stress on the tendon actuation system and reduced risk of unnecessary excitation of the flexible structure. Figure 6.20 compares the actual state variables with the observer estimates under the sinusoidal trajectories. Again, the observer tracks the state evolution closely, confirming that the prescribed-time descriptor observer remains reliable beyond the step-response case.

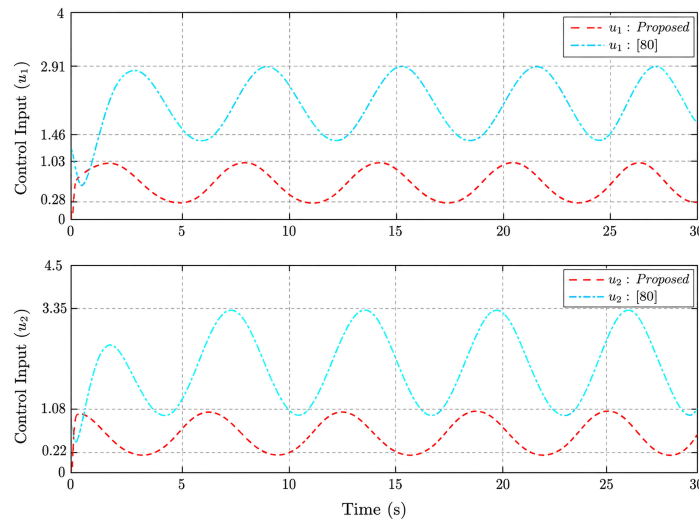


Figure 6.19 Comparison of control inputs under sinusoidal signal.

Taken together, the sinusoidal test demonstrates that the proposed controller maintains high-quality tracking not only for isolated changes in the reference, but also for continuously varying motions of the type expected in curved or tortuous surgical paths.

6.6.4 Multiple-Fault Resilience

The third test is designed to evaluate resilience under a more demanding combined-fault scenario. In this case, simultaneous actuator faults, sensor faults, and time-varying input delays are injected into the closed-loop system. The actuator fault and input delay are

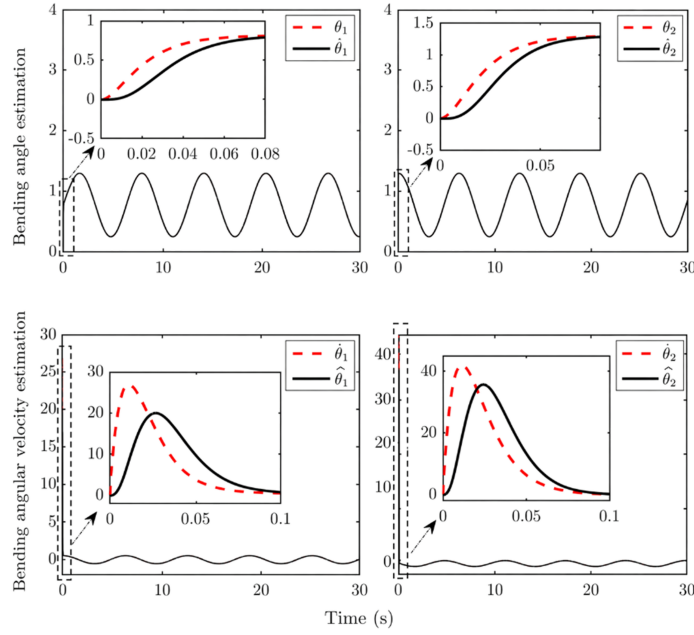


Figure 6.20 System states and observer estimates under sinusoidal references.

introduced as

$$f_{ai}(t) = \begin{cases} 0, & t < 20, \\ \frac{\pi}{15} + \frac{\pi}{30} \cos(3t), & t \geq 20, \end{cases} \quad i = 1, 2, \quad (6.76)$$

$$\tau_i(t) = 0.2 + 0.15 \sin(0.5t), \quad i = 1, 2. \quad (6.77)$$

The actuator-fault distribution matrix is selected as $\mathbf{G}_{fa} = [\mathbf{0}_{2 \times 2} \quad \mathbf{I}_2]^T$. To stress the output channel as well, the sensor fault is defined as

$$f_{si}(t) = \begin{cases} 0, & t < 35, \\ \frac{\pi}{30} \sin(0.5\pi t), & t \geq 35, \end{cases} \quad i = 1, 2, \quad (6.78)$$

with the corresponding sensor-fault distribution matrix $\mathbf{S}_{fs}$. The resulting tracking responses are reported in Figure 6.21. Even after the actuator fault is injected, the proposed controller recovers rapidly and returns to accurate tracking within a short interval. In the reported simulations, the recovery time is approximately 0.2 s, whereas the comparison method in [82] requires around 0.4 s and exhibits more pronounced oscillatory behavior. This indicates that the proposed architecture achieves not only fault accommodation, but also faster transient restoration.

Figure 6.22 compares the true states and the observer-based estimates under the same combined-fault scenario. The agreement remains close even after the sensor-side corruption

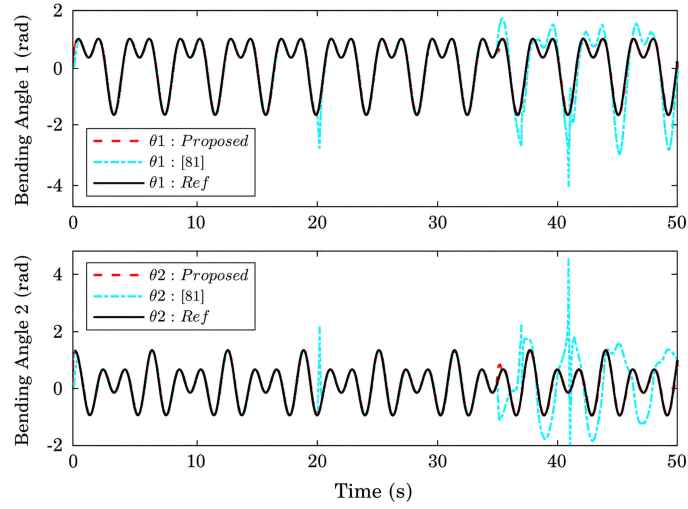


Figure 6.21 Comparison of continuum robot bending angles tracking under multiple faults.

becomes active, which confirms that the observer is able to reconstruct the true system behavior from faulty measurements. Figure 6.23 shows the sensor-fault estimation results. The estimated fault signals follow the injected fault pattern closely, demonstrating that the observer is not only reconstructing the state but also identifying the output-channel corruption with useful accuracy.

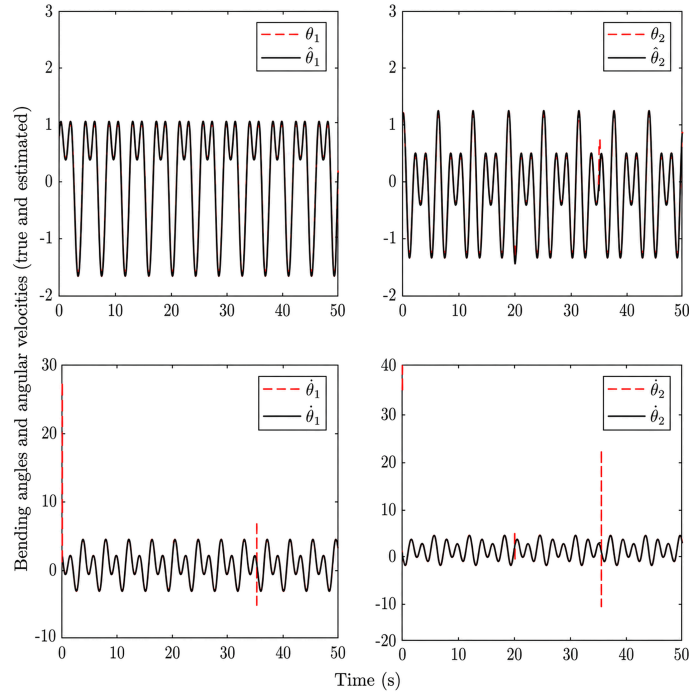


Figure 6.22 System states and observer estimates under multiple faults.

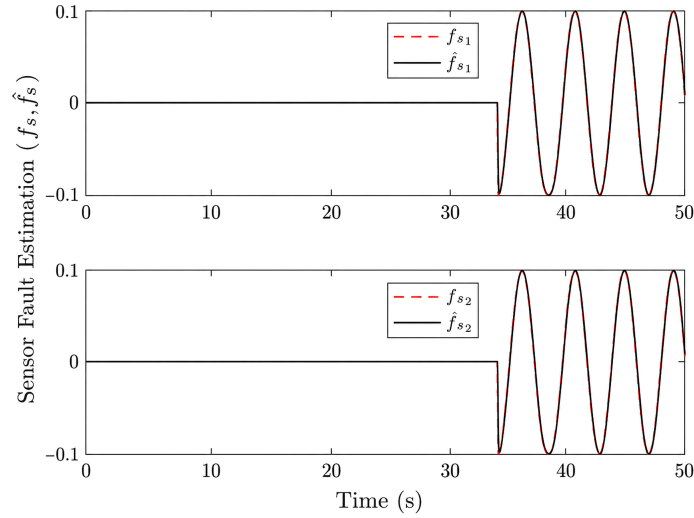


Figure 6.23 Sensor faults estimation.

This test is particularly important from the viewpoint of surgical robotics. In a realistic intervention, faults and delays rarely occur in isolation. Instead, sensing degradation, actuator-side perturbations, and communication or processing delays may overlap. The proposed method is able to preserve stable tracking in exactly this kind of composite setting, which supports the practical relevance of the design.

6.7 Chapter Summary

This chapter addressed the problem of safe and accurate execution for the two-segment tendon-driven continuum robot after the planning stage developed in Chapter 5. While the previous chapter focused on generating feasible and dexterity-aware references, the present chapter concentrated on the control layer required to track those references under both nominal and non-ideal operating conditions.

First, a control-oriented reduced-order model was introduced for the bending dynamics of the robot under the fixed-orientation assumption. On this basis, a prescribed-time safe-tracking framework was developed by combining monotone tube boundaries with a transformed-error sliding structure. This formulation converted the constrained tracking problem into a bounded stabilization problem while preserving explicit transient safety requirements. An actor-critic learning term was then incorporated to improve control efficiency without altering the stabilizing core of the prescribed-time design. The nominal simulation results showed that this controller achieves faster convergence, keeps the tracking error inside

the prescribed tube, and reduces the required control effort with respect to the benchmark strategy.

The chapter then extended the nominal design to a more realistic observer-based fault-tolerant framework. To account for output corruption, actuator faults, sensor faults, time-varying input and output delays, saturation, disturbances, and residual nonlinear uncertainty, the fault-affected model was rewritten in augmented form and a prescribed-time descriptor observer was introduced. This observer reconstructs both the robot states and the output-channel corruption term within a prescribed time. Based on these estimates, a composite deep-learning-based fault-tolerant control law was formulated by combining baseline feedback, prescribed-time shaping, an auxiliary saturation compensator, and an online DNN approximation of the residual optimal control component. Unlike the nominal monotone-tube formulation, the fault-tolerant design guarantees prescribed-time convergence of the augmented tracking error to an adjustable compact residual set under non-ideal operating conditions.

Finally, numerical studies under step, sinusoidal, and multiple-fault scenarios showed that the proposed framework preserves accurate tracking and stable closed-loop behaviour even when several non-ideal effects act simultaneously. The observer provided rapid state and fault reconstruction, while the control law maintained smaller tracking errors and lower control effort than the benchmark methods. Taken together, the developments of this chapter completed the transition from feasible motion generation to nominal tube-based safe tracking and prescribed-time fault-tolerant execution for the studied TDCR architecture. In this way, the chapter closes the gap between feasibility-aware planning and robust physical execution, which is a central requirement for the practical deployment of tendon-driven continuum robots in constrained and safety-critical environments.

Chapter 7

Conclusions and Further Research

7.1 Conclusions

Compliant robotic systems for human-centred healthcare present a common set of design challenges. Mechanisms must remain compatible with the human body, robot configurations must be feasible under physical constraints and loading conditions, and control strategies must ensure safe and accurate execution despite uncertainty and practical non-idealities. This thesis addressed these challenges across two complementary domains: wearable hand rehabilitation devices and tendon-driven continuum robots. The work combined mechanism design, experimental characterisation, model-based optimisation, learning-assisted planning, capability mapping, and robust control.

Chapter 3 presented a hybrid wearable system for hand rehabilitation, combining an RCM-based finger exoskeleton with glove-integrated 3D-printed inflatable chambers. The RCM linkage was designed to guide MCP and PIP flexion–extension through virtual centres of rotation located close to the anatomical joint regions, while keeping the physical joints of the mechanism outside the finger joint axes. This design choice was intended to reduce misalignment and relative motion at the human-device interface. The pneumatic resistance layer was developed through iterative prototyping, from early Flexojoint geometries to a compact pillow-shaped chamber with improved contact stability and sealing reliability.

The experimental characterisation showed that the internal pressure, reaction force, and resistive torque generally increase with bending angle and initial inflation pressure. For an initial pressure of 1.5 bar, the forward configuration generated a peak torque of approximately 0.42 Nm at 45° of flexion, while the backward configuration reached approximately 0.30 Nm under the same conditions. The resulting nonlinear torque–angle behaviour indicates that the proposed pneumatic layer can provide adjustable resistance for different rehabilitation stages.

However, this validation remains limited to quasi-static mechanical characterisation and does not yet demonstrate clinical effectiveness.

Chapter 4 established the model-based foundation for the TDCR part of the thesis. A discretised PCC kineto-static formulation was developed to account for tendon forces, gravity, vertebra-level friction, backbone elasticity, and external tip loads in a two-segment vertebra-based continuum robot. This framework was used to analyse the feasible static workspace, distinguishing geometric reachability from configurations that can actually be maintained in static equilibrium under admissible tendon forces.

The same framework was then extended to the dimensional synthesis of tendon-routing parameters. A bi-objective optimisation problem was formulated to study the trade-off between feasible static workspace and mechanical effort. The results showed that tendon-routing radii and inter-segment angular offset have a direct influence on static reachability and actuation efficiency. In particular, the analysis indicated that larger angular offsets improve the consistency of feasible workspace performance, with the prototype-oriented design converging toward tendon-routing values around $r_1 = 21$ mm, $r_2 = 22.5$ mm, and $\alpha = 45^\circ$.

Chapter 5 addressed planning for tendon-driven continuum robots through two complementary strategies. The first strategy combined supervised learning with constrained optimisation for task-space inverse kinematics. The neural network provided fast initial estimates, while the optimisation step refined the solution and enforced physical constraints. This hybrid approach improved accuracy and feasibility compared with purely learning-based prediction, but it also introduced online computational cost.

To reduce the need for online optimisation, an offline capability-map framework was developed. The capability map stores feasible voxel-orientation samples together with their configurations, pose residuals, and dexterity indicators. Runtime configuration selection was then formulated as a database-driven bi-objective problem using Pareto ranking and ideal-point selection. This allowed the robot to choose configurations that balance pose accuracy and manipulability without solving a full inverse-kinematics problem online. The experimental validation on the TDCR prototype showed the practical potential of the database-driven approach.

Chapter 6 addressed the control layer required for safe execution. A prescribed-time safe-tracking framework was developed using monotone tube boundaries and transformed tracking errors. The objective was to keep the tracking error within predefined bounds and guarantee convergence within a prescribed time. The fault-tolerant extension incorporated observer-based estimation and learning-assisted compensation to address uncertainties, actu-

ator faults, sensor faults, delays, and input saturation. The simulation results showed that the proposed control framework can improve tracking performance and robustness under non-ideal operating conditions.

Overall, the thesis shows that compliant robotic systems for healthcare should not be designed only from the viewpoint of motion generation. Their practical usefulness depends on anatomical compatibility, mechanical feasibility, constrained planning, and safe control. The transition from wearable rehabilitation mechanics to continuum-robot planning and control reflects this common concern: ensuring that compliant systems remain predictable, adjustable, and safe under practical human-centred constraints.

7.2 Further Research

Several research directions remain open. For the rehabilitation system, future work should focus on improving the mechanical integration of the RCM linkage and pneumatic chambers, reducing bulk, and increasing comfort during repeated use. Closed-loop pressure regulation should be added to adjust the resistive torque during a rehabilitation session according to user effort, joint angle, or therapy stage. Further tests with healthy participants and patients are also required to evaluate comfort, repeatability, donning and doffing, long-term durability, and clinical relevance.

For the TDCR modelling and design framework, future research should include more detailed physical effects such as tendon hysteresis, backlash, distributed friction, material nonlinearities, and manufacturing tolerances. The kineto-static predictions should also be validated experimentally by comparing simulated and measured tip poses under controlled external loading. The dimensional synthesis framework could be extended to three-segment robots and to additional constraints related to miniaturisation, sterilisation, and surgical deployment.

For the capability-map framework, future work should move from single-pose configuration selection to trajectory-level planning. This would require connecting neighbouring feasible configurations while preserving tendon-length continuity, dexterity, and actuation feasibility. The database could also be updated using experimental feedback, allowing the capability map to become more consistent with the real robot over time.

For the control part, the prescribed-time and fault-tolerant strategies should be implemented on the physical TDCR prototype. Hardware experiments should include sensor noise, actuator saturation, tendon friction, delays, and injected faults. A long-term direction is to connect the offline capability-map planner with the prescribed-time controller, so that

feasible configurations selected from the database can be executed safely and robustly in real time.

In conclusion, this thesis provides a foundation for compliant robotic systems that combine mechanical design, feasibility-aware planning, and safe control. Future developments should continue this integration and move from prototype-level and simulation-based validation toward more complete experimental and clinical evaluation.

References

- [1] Michelle E. Mlinac and Michelle C. Feng. Assessment of activities of daily living, self-care, and independence. *Archives of Clinical Neuropsychology*, 31(6):506–516, 2016.
- [2] Merja Rantakokko, Minna Mänty, and Taina Rantanen. Mobility decline in old age. *Exercise and Sport Sciences Reviews*, 41(1):19–25, 2013.
- [3] Ilse Lamers, Anneleen Maris, Deborah Severijns, Wouter Dielkens, Sander Geurts, Bart Van Wijmeersch, and Peter Feys. Upper limb rehabilitation in people with multiple sclerosis: a systematic review. *Neurorehabilitation and Neural Repair*, 30(8):773–793, 2016.
- [4] Li-Chieh Kuo, Shih-Wei Chen, Chien-Ju Lin, Wei-Jr Lin, Sheng-Che Lin, and Fong-Chin Su. The force synergy of human digits in static and dynamic cylindrical grasps. *PLoS One*, 8(3):e60509, 2013.
- [5] Alain Favetto. *Glove exoskeleton for extra-vehicular activities: analysis of requirements and prototype design*. PhD thesis, Politecnico di Torino, 2014. Doctoral thesis, 197 pages.
- [6] Mohammad Jabari, Giovanni Colucci, Simone Duretto, Carmen Visconte, and Giuseppe Quaglia. A 3d printed wearable glove with inflatable chambers. In Giuseppe Quaglia, Giovanni Boschetti, and Giuseppe Carbone, editors, *Advances in Italian Mechanism Science*, volume 164 of *Mechanisms and Machine Science*, pages 99–107. Springer, Cham, 2024.
- [7] Catherine E. Lang, Marghuretta D. Bland, Ryan R. Bailey, Sydney Y. Schaefer, and Rebecca L. Birkenmeier. Assessment of upper extremity impairment, function, and activity after stroke: foundations for clinical decision making. *Journal of Hand Therapy*, 26(2):104–115, 2013.
- [8] Zahra Jiryaee and Amir Salar Jafarpisheh. The usefulness of assistive soft robotics in the rehabilitation of patients with hand impairment: a systematic review. *Journal of Bodywork and Movement Therapies*, 39:398–409, 2024.
- [9] Barry Trimmer. Soft robots. *Current Biology*, 23(15):R639–R641, 2013.
- [10] N. S. K. Ho, K. Y. Tong, X. L. Hu, K. L. Fung, X. J. Wei, W. Rong, and E. A. Susanto. An emg-driven exoskeleton hand robotic training device on chronic stroke subjects: task training system for stroke rehabilitation. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–5. IEEE, 2011.

- [11] Hyunki In, Brian Byunghyun Kang, Min Ki Sin, and Kyu Jin Cho. Exo-glove: a wearable robot for the hand with a soft tendon routing system. *IEEE Robotics & Automation Magazine*, 22(1):97–105, 2015.
- [12] Panagiotis Polygerinos, Kevin C. Galloway, Siddharth Sanan, Maxwell Herman, and Conor J. Walsh. Emg controlled soft robotic glove for assistance during activities of daily living. In *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 55–60. IEEE, 2015. Article 7281175.
- [13] Hong Kai Yap, Jeong Hoon Lim, Fatima Nasrallah, Fan-Zhe Low, James C. H. Goh, and Raye C. H. Yeow. Mrc-glove: a fmri compatible soft robotic glove for hand rehabilitation application. In *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 735–740. IEEE, 2015.
- [14] Lisen Ge, Feifei Chen, Dong Wang, Yifan Zhang, Dong Han, Tao Wang, and Guoying Gu. Design, modeling, and evaluation of fabric-based pneumatic actuators for soft wearable assistive gloves. *Soft Robotics*, 7(5):583–596, 2020.
- [15] Leonardo Cappello, Jan T. Meyer, Kevin C. Galloway, Jeffrey D. Peisner, Rachael Granberry, Diana A. Wagner, Sven Engelhardt, Sabrina Paganoni, and Conor J. Walsh. Assisting hand function after spinal cord injury with a fabric-based soft robotic glove. *Journal of NeuroEngineering and Rehabilitation*, 15:59, 2018.
- [16] Jianwei Lai, Aiguo Song, Jiajin Wang, Ye Lu, Ting Wu, Huijun Li, Baoguo Xu, and Xiangshan Wei. A novel soft glove utilizing honeycomb pneumatic actuators (hpas) for assisting activities of daily living. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:3223–3233, 2023.
- [17] Mohammad Jabari, Carmen Visconte, Giuseppe Quaglia, and Med Amine Laribi. Multi-objective optimization for dimensional synthesis of tendon placement and structural design for energy-efficient and feasible static workspace in continuum robots. *Robotics and Autonomous Systems*, 197:105300, 2026.
- [18] Mohammad Jabari, Carmen Visconte, Giuseppe Quaglia, and Med Amine Laribi. Feasible static workspace optimization of tendon driven continuum robot based on euclidean norm. In Med Amine Laribi, Giuseppe Carbone, Doina Pisla, and Saïd Zeghloul, editors, *New Trends in Medical and Service Robotics*, volume 186 of *Mechanisms and Machine Science*, pages 489–501. Springer, Cham, 2025.
- [19] Sahar Refaat Habib and Nashwa Nabil Kamal. Stature estimation from hand and phalanges lengths of egyptians. *Journal of Forensic and Legal Medicine*, 17(3):156–160, 2010.
- [20] Robert J. III Webster and Bryan A. Jones. Design and kinematic modeling of constant curvature continuum robots: a review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.
- [21] Jessica Burgner-Kahrs, D. Caleb Rucker, and Howie Choset. Continuum robots for medical applications: a survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

- [22] Georg Böttcher, Sven Lilge, and Jessica Burgner-Kahrs. Design of a reconfigurable parallel continuum robot with tendon-actuated kinematic chains. *IEEE Robotics and Automation Letters*, 6(2):1272–1279, 2021.
- [23] Yasaman Pedari, Aida Parvaresh, and S. Ali A. Moosavian. Spatial shape estimation of a tendon-driven continuum robotic arm using a vision-based algorithm. In *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, pages 625–630. IEEE, 2019.
- [24] Sven Lilge and Jessica Burgner-Kahrs. Kinetostatic modeling of tendon-driven parallel continuum robots. *IEEE Transactions on Robotics*, 39(2):1563–1579, 2023.
- [25] Shifa Sulaiman, Francesco Schetter, Mohammad Gohari, and Fanny Ficuciello. An adaptive neuro-controller developed for a prosthetic hand wrist. arXiv preprint arXiv:2510.19068, 2025.
- [26] Christian Muhmann, Reinhard M. Grassmann, Max Bartholdt, and Jessica Burgner-Kahrs. Toward dynamic control of tendon-driven continuum robots using clarke transform. arXiv preprint arXiv:2503.20693, 2025.
- [27] Lorenzo Toscano, Gabriele Cazzulani, and Francesco Braghin. Forward kinematic modeling of constant curvature continuum robots using dual quaternions. In *7th ECCOMAS Thematic Conference on Smart Structures and Materials*, page 16, Ponta Delgada, Portugal, 2015.
- [28] Carlo Alessi, Camilla Agabiti, Daniele Caradonna, Cecilia Laschi, Federico Renda, and Egidio Falotico. Rod models in continuum and soft robot control: a review. arXiv preprint arXiv:2407.05886, 2024.
- [29] Han Yuan, Lili Zhou, and Wenfu Xu. A comprehensive static model of cable-driven multi-section continuum robots considering friction effect. *Mechanism and Machine Theory*, 135:130–149, 2019.
- [30] Zheng Li, Ruxu Du, Haoyong Yu, and Hongliang Ren. Statics modeling of an underactuated wire-driven flexible robotic arm. In *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 326–331. IEEE, 2014.
- [31] Wisama Khalil. Dynamic modeling of robots using recursive newton-euler techniques. In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Funchal, Portugal, 2010.
- [32] Elie Gautreau, Xavier Bonnet, and Med Amine Laribi. Instrumented bio-inspired cable-driven compliant continuum robot: static modeling and experimental evaluation. *Robotica*, 42(9):2924–2948, 2024.
- [33] Matthias Tummers. *Cosserat rod modeling of continuum robots: application to concentric agonist-antagonist robots*. PhD thesis, Université Grenoble Alpes, 2023.
- [34] Andrea Gotelli. *Dynamics modelling of continuum parallel robots*. PhD thesis, École Centrale de Nantes, 2024.

- [35] Xiang Wang, Y. Ding, L. Zeng, C. Zhu, B. Wu, and Kai Xu. Kinetostatic modeling of continuum delta robot with variable curvature continuum joints. *Journal of Mechanisms and Robotics*, 15(3):031005, 2023.
- [36] Peiyi Wang, Daniel Feliu-Talegon, Yuchen Sun, Z. Xie, W. Xin, M. S. Nazeer, C. Della Santina, Cecilia Laschi, and Federico Renda. Strain-based shape and 3-d force estimation for rod-driven continuum robots with stretch sensors. *IEEE Transactions on Robotics*, 42:894–911, 2026.
- [37] Elie Gautreau, Xavier Bonnet, and Med Amine Laribi. Optimal synthesis and experimental validation of a bio-inspired variable stiffness universal compliant joint for continuum robots. In *IFTToMM World Congress on Mechanism and Machine Science*, pages 419–428. Springer Nature Switzerland, Cham, 2023.
- [38] Xinyu Li, Qian Tang, Guoxin Yin, Gang Zheng, Jessica Burgner-Kahrs, Cesare Stefanini, and Ke Wu. Collaborative continuum robots: a survey. arXiv preprint arXiv:2601.10721, 2026.
- [39] Han Yuan and Zheng Li. Workspace analysis of cable-driven continuum manipulators based on static model. *Robotics and Computer-Integrated Manufacturing*, 49:240–252, 2018.
- [40] Maria Paula Huertas Niño, Mohamed Boutayeb, and Dominique Martinez. A hybrid tendon-driven continuum robot that avoids torsion under external load. *Frontiers in Robotics and AI*, 12:1576209, 2025.
- [41] Botao Lin, Shuang Song, and Jiaole Wang. Reconfigurable tendon-driven robots: eliminating inter-segmental coupling via independently lockable joints. arXiv preprint arXiv:2507.17163, 2025.
- [42] Francesco Stella, Josie Hughes, Daniela Rus, and Cosimo Della Santina. Prescribing cartesian stiffness of soft robots by co-optimization of shape and segment-level stiffness. *Soft Robotics*, 10(4):701–712, 2023.
- [43] Dong Liang, Yi Mao, Yimin Song, and Tao Sun. Kinematics, dynamics and multi-objective optimization based on singularity-free task workspace for a novel scara parallel manipulator. *Journal of Mechanical Science and Technology*, 38:423–438, 2024.
- [44] Ines Ben Hamida, Med Amine Laribi, Abdelfattah Mlika, Lotfi Romdhane, Saïd Zeghloul, and Giuseppe Carbone. Multi-objective optimal design of a cable driven parallel robot for rehabilitation tasks. *Mechanism and Machine Theory*, 156:104141, 2021.
- [45] Jens Reinecke, Bastian Deutschmann, Alexander Dietrich, Simon R. Eugster, and Marco Hutter. A computational approach for internal tendon routing channels in a tendon-driven continuum joint. *Soft Robotics*, 12(2):280–289, 2025.
- [46] Zhiwei Cui, Kemeng Du, Ligang Jin, Rong Song, Yu Li, and Fumin Sun. Multi-objective dimension and shape optimization design of the cable-driven parallel robot based on the response surface. *Electronics*, 15(2):315, 2026.

- [47] Fahad Iqbal, Mojtaba Esfandiari, Golchehr Amirkhani, Hamidreza Hoshyarmansh, Sanju Lama, Mahdi Tavakoli, and Garnette R. Sutherland. Continuum and soft robots in minimally invasive surgery: a systematic review. *IEEE Access*, 13:24053–24079, 2025.
- [48] Tomas da Veiga, James H. Chandler, Peter Lloyd, Giovanni Pittiglio, Nathan J. Wilkinson, Ali K. Hoshier, Russell A. Harris, and Pietro Valdastri. Challenges of continuum robots in clinical context: a review. *Progress in Biomedical Engineering*, 2(3):032003, 2020.
- [49] Priyanka Rao, Quentin Peyron, Sven Lilge, and Jessica Burgner-Kahrs. How to model tendon-driven continuum robots and benchmark modelling performance. *Frontiers in Robotics and AI*, 7:630245, 2021.
- [50] Pierre E. Dupont, Nabil Simaan, Howie Choset, and D. Caleb Rucker. Continuum robots for medical interventions. *Proceedings of the IEEE*, 110(7):847–870, 2022.
- [51] Matthias Tummers, Vincent Lebastard, Frédéric Boyer, Jocelyne Troccaz, Barbara Rosa, and M. Taha Chikhaoui. Cosserat rod modeling of continuum robots from newtonian and lagrangian perspectives. *IEEE Transactions on Robotics*, 39(3):2360–2378, 2023.
- [52] Azadeh Doroudchi, Zhi Qiao, Wenlong Zhang, and Spring Berman. Implementation of a cosserat rod-based configuration tracking controller on a multi-segment soft robotic arm. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 620–626. IEEE, 2023.
- [53] Yifan Wang, Zhonghao Wu, Longfei Wang, Bo Feng, and Kai Xu. Inverse kinematics and dexterous workspace formulation for 2-segment continuum robots with inextensible segments. *IEEE Robotics and Automation Letters*, 7(1):510–517, 2022.
- [54] Amir Trabelsi, Juan Sandoval, Abdelfattah Mlika, Samir Lahouar, Said Zeghloul, and Med Amine Laribi. Robot base placement and tool mounting optimization based on capability map for robot-assistant camera holder. *Robotica*, 42(8):2489–2510, 2024.
- [55] Amir Trabelsi, Juan Sandoval, Abdelfattah Mlika, Samir Lahouar, Said Zeghloul, J. Cau, and Med Amine Laribi. Optimal multi-robot placement based on capability map for medical applications. In *Advances in Service and Industrial Robotics*, pages 333–342. Springer, Cham, 2022.
- [56] Ibrahim A. Seleem, Haitham El-Hussieny, and Hiroyuki Ishii. Imitation-based motion planning and control of a multi-section continuum robot interacting with the environment. *IEEE Robotics and Automation Letters*, 8(3):1351–1358, 2023.
- [57] Jonathan Külz, Sehoon Ha, and Matthias Althoff. A design co-pilot for task-tailored manipulators. arXiv preprint arXiv:2509.13077, 2025.
- [58] Shengge Cao, Jingjun Yu, Jie Pan, and X. Pei. Design and moving capability analysis of a flexible continuum robot with rolling contact. *Journal of Mechanical Engineering*, 57(19):21–29, 2021.

- [59] Duc Tu Vu, N. A. Phan, S. T. Ngo, M. T. Phan, T. A. Truong, C. C. Nguyen, P. T. Phan, H. P. Phan, T. N. Do, and M. T. Thai. Soft robotics and advanced technologies for minimally invasive bioprinting: the future of internal organ repair. *Advanced Science*, page e23532, 2026.
- [60] Turhan Can Kargin and Jakub Kołota. A reinforcement learning approach for continuum robot control. *Journal of Intelligent & Robotic Systems*, 109:77, 2023.
- [61] Abdelhamid Ghoul, Selman Djeflal, Hao Wang, Kamel Kara, and Mohamed Laid Hadjili. Enhancing surgical robotics: a dynamic model and optimized control strategy for cable-driven continuum robots. *Journal of Mechanisms and Robotics*, 17(2):021002, 2025.
- [62] Jiewen Lai, Kaicheng Huang, Bo Lu, Qingxiang Zhao, and Henry K. Chu. Verticalized-tip trajectory tracking of a 3d-printable soft continuum robot: enabling surgical blood suction automation. *IEEE/ASME Transactions on Mechatronics*, 27(3):1545–1556, 2022.
- [63] Ziwen Wang, Teng Wang, Baoliang Zhao, Yucheng He, Ying Hu, Bing Li, Peng Zhang, and Max Q.-H. Meng. Hybrid adaptive control strategy for continuum surgical robot under external load. *IEEE Robotics and Automation Letters*, 6(2):1407–1414, 2021.
- [64] Matteo Russo, S. M. Hadi Sadati, Xin Dong, Abdelkhalick Mohammad, Ian D. Walker, Christos Bergeles, Kai Xu, and Dragos A. Axinte. Continuum robots: an overview. *Advanced Intelligent Systems*, 5(5):2200367, 2023.
- [65] Mohammad Jabari, Andrea Botta, Luigi Tagliavini, Carmen Visconte, and Giuseppe Quaglia. A safe, high-precision reinforcement learning-based optimal control of surgical continuum robots: a monotone tube boundary approach with prescribed-time control capability. *Robotics and Autonomous Systems*, 190:104992, 2025.
- [66] Abdelhamid Ghoul, Kamel Kara, Mohamed Benrabah, and Mohamed Laid Hadjili. Optimized nonlinear sliding mode control of a continuum robot manipulator. *Journal of Control, Automation and Electrical Systems*, 33(5):1355–1363, 2022.
- [67] Abdelhamid Ghoul, Kamel Kara, Mohamed Benrabah, and B. Nasri. Control of continuum robot using two optimized pid controller. In *Multi Conference on Electrical Engineering (CEE)*, pages 1–6, 2021.
- [68] Yakang Wang, Yuzhe Qian, and Weipeng Liu. A novel model predictive control strategy for continuum robot: optimization and application. In *Advances in Applied Nonlinear Dynamics, Vibration, and Control*, volume 1152 of *Lecture Notes in Electrical Engineering*, pages 484–497. Springer Nature Singapore, 2024.
- [69] Yuhua Song, Lifeng Zhu, Jinfeng Li, Jiawei Deng, Cheng Wang, and Aiguo Song. Mpc design of a continuum robot for pulmonary interventional surgery using koopman operators. *IEEE Robotics and Automation Letters*, 9(12):10819–10826, 2024.
- [70] Hexiang Yuan and Juanjuan Zhang. Closed-loop trajectory tracking control for continuum robots: an mpc-based approach. In *2025 10th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 723–727. IEEE, 2025.

- [71] Xu Liang, Guoqing He, Tian Su, Wei Wang, Chao Huang, Qingxiang Zhao, and Zeng-Guang Hou. Finite-time observer-based variable impedance control of cable-driven continuum manipulators. *IEEE Transactions on Human-Machine Systems*, 52(1):26–40, 2022.
- [72] Yi Shi, Bowen Yi, Wei Xie, and Weidong Zhang. Enhancing prescribed performance of tracking control using monotone tube boundaries. *Automatica*, 159:111304, 2024.
- [73] Zhiliang Liu, Chong Lin, and Yun Shang. Prescribed-time adaptive neural feedback control for a class of nonlinear systems. *Neurocomputing*, 511:155–162, 2022.
- [74] Min Wang and Anle Yang. Dynamic learning from adaptive neural control of robot manipulators with prescribed performance. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(8):2244–2255, 2017.
- [75] Guoxing Wen and C. L. Philip Chen. Optimized backstepping consensus control using reinforcement learning for a class of nonlinear strict-feedback-dynamic multi-agent systems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(3):1524–1536, 2023.
- [76] Omkar Sudhir Patil, Duc M. Le, Max L. Greene, and Warren E. Dixon. Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network. *IEEE Control Systems Letters*, 6:1855–1860, 2022.
- [77] S. Mohammadreza Ebrahimi, Farid Norouzi, Hossein Dastres, Reza Faieghi, Mehdi Naderi, and Milad Malekzadeh. Sensor fault detection and compensation with performance prescription for robotic manipulators. *Journal of the Franklin Institute*, 361(7):106742, 2024.
- [78] Hongyun Liu, Weidong Liu, and Jinbo Zhong. Adaptive neural network fault-tolerant control of tendon actuated continuum robots under actuator failures and external disturbances. *Robotics and Autonomous Systems*, 198:105310, 2026.
- [79] Meng Ding, Hai Wu, Xianjie Zheng, and Yu Guo. Adaptive predefined-time attitude stabilization control of space continuum robot. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(2):647–651, 2024.
- [80] Mohammad Jabari, Carmen Visconte, Giuseppe Quaglia, and Med Amine Laribi. Prescribed-time fault-tolerant optimal control of continuum surgical robots via composite deep learning. *European Journal of Control*, page 101514, 2026.
- [81] Wenqi Liu, Zehua Ye, Dan Zhang, Jun Cheng, and Huaicheng Yan. Adaptive estimator-based nonsingular fast terminal sliding mode control of robotic manipulator systems under fdi attacks and actuator failure. *Journal of the Franklin Institute*, 361(15):107129, 2024.
- [82] Pengxin Yang, Shuang Zhang, Xinbo Yu, and Wei He. Reinforcement-learning-based finite-time fault-tolerant control for a manipulator with actuator faults. *IEEE Transactions on Cybernetics*, 55(6):2621–2632, 2025.