



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
MECCANICA E SCIENZE AVANZATE DELL'INGEGNERIA
Ciclo 38

Settore Concorsuale: 09/A2 - MECCANICA APPLICATA ALLE MACCHINE

Settore Scientifico Disciplinare: ING-IND/13 - MECCANICA APPLICATA ALLE MACCHINE

SELF-CALIBRATION OF CABLE-DRIVEN PARALLEL ROBOTS

Presentata da: Filippo Zoffoli

Coordinatore Dottorato

Lorenzo Donati

Supervisore

Edoardo Ida

Co-supervisore

Marco Carricato

Esame finale anno 2026

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SELF-CALIBRATION OF CABLE-DRIVEN PARALLEL ROBOTS

A dissertation submitted in partial fulfillment
of the requirements for the degree of Doctor of Philosophy
in Mechanics and Advanced Engineering Sciences

Filippo Zoffoli

Bologna, Italy,
April 1, 2026

*To Cristina, Gabriele,
Fede, Al and Sara.*



This work is licensed under the Creative Commons
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

To view a copy of this license, visit
<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

Abstract

Cable-Driven Parallel Robots (*CDPRs*) employ multiple cables arranged in parallel, attached to the End-Effector (*EE*), which is located in a rigid platform. The cable length variation, given by the winding and unwinding of the cables on servo winches, allows the *EE* motion to perform manipulation tasks. Thanks to their high payload-to-weight ratio, easy scalability to large dimensions, and good reconfiguration properties, *CDPRs* represent an effective solution for large-scale manipulation tasks, such as operations on building façades. In these applications, cable attachment points can be easily re-located to adapt the robot workspace to task-specific requirements, enabling versatile deployment with limited mechanical complexity.

Despite these advantages, *CDPRs* are affected by intrinsic challenges related to the unilateral nature of cables, which act as active constraints only with positive tensions. In overconstrained architectures, positive cable tensions can be ensured by exploiting actuation redundancy through hybrid joint-space position/force control strategies. While effective, these approaches are highly sensitive to modeling inaccuracies, sensor noise, and parameter uncertainties, conditions that are particularly relevant in reconfigurable systems and in unstructured environments such as construction sites. Moreover, every time the *CDPR* is reconfigured, the model parameters have to be recalibrated to achieve good motion accuracy. As *CDPRs* commonly rely on relative sensors, whose readings must be referenced at each system startup, accurate initial-pose (or initial sensor offset) estimation therefore becomes a critical requirement for reliable operation.

This thesis addresses these issues through two main research directions. The first focuses on workspace analysis for hybrid-controlled overconstrained *CDPRs*, with the goal of explicitly accounting for robustness with respect to control and measurement errors. A control-oriented workspace characterization is developed, leading to the definition of the Wrench-Feasible Error-Insensitive Workspace (*WFEIW*). This workspace represents the subset of the classical Wrench-Feasible Workspace in which user-defined accuracy requirements and admissible cable tension distributions are guaranteed even in the presence of bounded uncertainties and noise. The proposed framework provides a quantitative tool for evaluating the robustness of hybrid control strategies throughout the workspace, and can be applied as a criterion for optimal design and control strategy selection.

The second research direction concerns the self-calibration of *CDPRs*, with particular emphasis on the estimation of the initial end-effector pose using only internal sensors. The problem is investigated under conditions of limited observability and large sensor uncertainties, which are typical of reconfigurable and large-scale systems. Manual and automatic procedures are introduced to reduce data acquisition time for underactuated *CDPRs*. The impact of sensor noise and modeling uncertainties on calibration accuracy is analyzed, and a Monte Carlo framework is employed to assess the

influence of algorithmic and parametric choices on multiple performance indices. Finally, online self-calibration strategies are explored as an alternative to sequential offline approaches, aiming to further reduce overall calibration time while maintaining accuracy.

Contents

Abstract	7
1 Introduction	11
1.1 Robotics foundations	11
1.1.1 Classification of manipulators	12
1.1.2 Robot Applications	13
1.2 Cable-Driven Parallel Robots	15
1.3 Thesis Goals and Outline	21
2 Modeling	23
2.1 Geometric model	24
2.1.1 First-order kinematics	26
2.1.2 Second-order kinematics	27
2.2 Static model	28
2.3 Geometric model of deployable <i>CDPRs</i>	29
2.3.1 First-order kinematics of deployable <i>CDPRs</i>	31
2.3.2 Second-order kinematics for deployable <i>CDPRs</i>	31
2.4 Static model of deployable <i>CDPRs</i>	32
2.5 Workspace analysis	32
2.5.1 Overconstrained <i>CDPRs</i> and the Tension Distribution Problem . . .	33
2.5.2 Underactuated <i>CDPRs</i> and the Inverse Geometrico-Static Problem	33
2.5.3 Computational Implementation	34
3 Control-based workspace analysis of <i>OCDPRs</i>	37
3.1 Hybrid Control	38
3.2 Wrench-Feasible Error-Insensitive Workspace	39
3.2.1 <i>WFEIW</i> Analysis	41
3.2.2 Application examples	44
3.3 Design Optimization algorithm	48
4 Initial-Pose Self-Calibration methods for <i>CDPRs</i>	55
4.1 The initial-pose self-calibration problem	56
4.2 Data acquisition for <i>UACDPRs</i>	59
4.2.1 Manual method	59
4.2.1.1 Data acquisition procedure	60
4.2.1.2 Tentative solution generation	60
4.2.1.3 Experiments and results	60
4.2.2 Automatic methods	62
4.2.2.1 Selection of the measurement poses	64

4.2.2.2	Data-acquisition procedure	65
4.2.2.3	Initial guess	67
4.2.2.4	Experiments and results	67
4.3	Algorithm and parameter analysis	71
4.3.1	Preliminary analysis on parameter selection for deployable <i>OCDPRs</i>	72
4.3.1.1	Simulation Results	73
4.3.2	Measurement Pose Selection	75
4.3.3	Analysis of Parameter Influence	76
4.3.4	Results	81
4.3.5	Experimental validation	89
4.4	Online initial-pose self-calibration	94
4.4.1	Nested method	94
4.4.2	Recursive Least Squares	95
4.4.2.1	Unscented version	98
4.4.3	Extended Kalman Filter	99
4.4.3.1	Unscented version	101
4.4.4	Results	103
5	Conclusions	109
5.1	Main results	109
5.2	Open Issues	111
	Appendices	113
A	Proofs	115
A.1	Computation of the swivel angle	115
A.2	Computation of the tangency angle	115
A.3	Computation of the swivel angle pose derivative	116
A.4	Computation of the tangency angle pose derivative	116
A.5	Computation of the cable length pose derivative	117
A.6	Computation of the swivel angle second-order pose derivative	118
A.7	Computation of the cable length second-order pose derivative	118
A.8	Computation of the swivel angle pose derivative for deployable <i>CDPRs</i>	119
A.9	Computation of the tangency angle pose derivative for deployable <i>CDPRs</i>	120
A.10	Computation of the cable length pose derivative for deployable <i>CDPRs</i>	120
A.11	Computation of the cable length second-order pose derivative for deployable <i>CDPRs</i>	121
	List of Author Publications	123

Chapter 1

Introduction

Following the ISO 8373 standard on robotics vocabulary, a robot is a *programmed actuated mechanism with a degree of autonomy to perform locomotion, manipulation or positioning*. This definition encompasses a wide range of complex systems, which, in turn, are generally composed of multiple subsystems (such as mechanical, control, electric, actuation, and sensory systems). Therefore, robotics is intrinsically a multi-disciplinary field in which disciplines such as computer science, automatic control, mechanics, electronics, and electrical engineering (among others) converge to study the intelligent interplay between perception and action [1].

The key features that differentiate a robot from a "classical" machine are flexibility, thus the ability to execute different tasks, and a degree of autonomy in decision-making and environmental interaction. To control the robot's motion effectively, the conventional approach relies on physical modeling to predict system behavior. This model (usually kinematic and/or dynamic) relies on parameters, treated as constant variables, which must be precisely estimated to achieve high-performance metrics such as speed, acceleration, and positioning accuracy within large workspaces.

In the following, some concepts that will form the basis for the subsequent sections are introduced (Sec. 1.1). After a general classification of manipulators (Sec. 1.1.1), the most common robotic applications are presented (Sec. 1.1.2). Then, Cable-Driven Parallel Robots (CDPRs) are introduced in Sec. 1.2, together with a way of classifying them and a presentation of their strong points and problems. Finally, the thesis objectives and outline are defined in Sec. 1.3

1.1 Robotics foundations

The first appearance of robots is attributed to the late 1960s when teleoperated servoed arms were developed to help humans in hazardous operations [2]. Then, technological progress in the second half of the twentieth century, with the development of integrated circuits, digital computers, and microelectronic components, enabled the design and programming of computer-controlled robots. The mass production of goods drove the adoption of robots in industry, where high volumes, harsh environments, and precision were required. Currently, robots find applications not only in the industrial world but in many other fields, including cleaning, search and rescue, underwater, space, flight, autonomous drive, and medical applications. The main trend of recent years is the use of modern artificial intelligence techniques and hardware-related breakthroughs to improve the autonomy capabilities in poorly structured en-

vironments, where few boundary conditions are defined a priori.

From a practical standpoint, a robot is usually composed of four essential components [3]:

1. the mechanical system, which comprises the actuation elements, the links, and the interfaces with the external environment;
2. the power system, which includes the elements that manage the power supply for the actuation system, allowing the robot, together with the mechanical system, to exert an action;
3. the sensing system, which includes the sensors and the signal conditioning for enabling the "perception" capability;
4. the control system, which, acting as a bridge between perception and action, allows for signal processing and elaboration, motion control, autonomous thinking, and human-machine interface. It is implemented in proprietary, partially open, or open software.

A macro classification of robots may be done based on the main task and motion capabilities they possess:

Manipulators have a fixed base and their main task regards object manipulation or transformation;

Mobile robots have a mobile base and are specialized in transport, monitoring, or inspection tasks.

Although in the following we will focus only on manipulators, it is necessary to note that this is a weak boundary, and several architectures have been proposed in the past to combine the advantages of both mobile and manipulator robots.

Among the mechanical elements of a manipulator, we define the End-Effector (*EE*) as the device that manipulates, monitors, interacts with, or transforms objects. It constitutes the interface between the robot and the external environment. Links and joints, which can be distinguished between actuated and passive, connect the fixed base to the *EE* [4].

1.1.1 Classification of manipulators

A classification can be made based on how the links and joints are arranged with respect to the *EE* and the fixed base (Fig. 1.1). This categorization is particularly relevant for building the robot's physical model, which is usually composed of kinematic and dynamic models, both of which are necessary for the control system. A manipulator is called:

Serial when the *EE* is connected to the fixed base only through a single open kinematic chain. Each kinematic pair is typically revolute or prismatic and is actuated by a motor;

Parallel when multiple closed kinematic chains, composed of links and joints, are interposed between the *EE* and the fixed base. In this case, the motors can be positioned on the base, as not all joints have to be actuated.

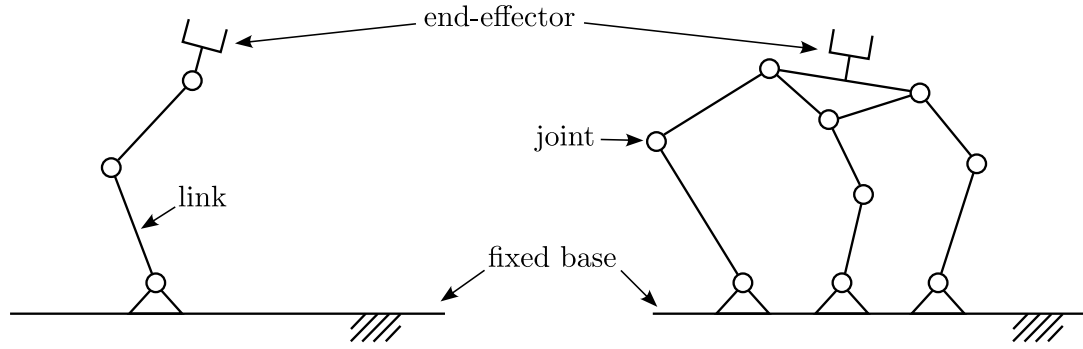
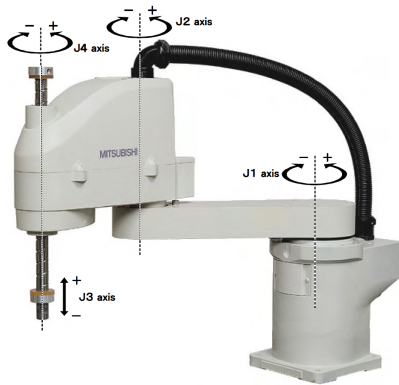


Figure 1.1: Serial (left) and parallel (right) robot architecture: the *EE* is connected to the base via an open chain (serial) or via multiple closed chains (parallel) composed of links and joints.



(a) Scara manipulator (Source: Wikipedia).



(b) Stewart platform (Source: Acrome).

Figure 1.2: Examples of robot manipulators: (a) serial and (b) parallel.

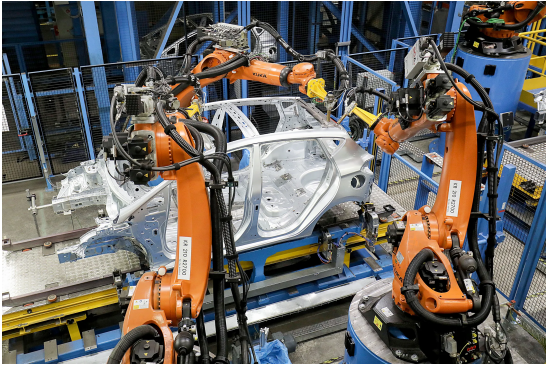
Serial manipulators, such as the well established Scara (Fig. 1.2a), generally have greater dexterity and larger reachable workspaces than parallel manipulators, but lower payloads and accuracy, for the same installation space [5]. While parallel architectures, such as the Stewart platform (Fig. 1.2b), have a higher payload-to-power ratio due to their greater rigidity, which allows for better dynamics features, and are more precise due to higher structural stiffness. On the other hand, their reachable workspace is usually smaller, mainly due to the actuators' limited strokes and possible interference between adjacent legs.

1.1.2 Robot Applications

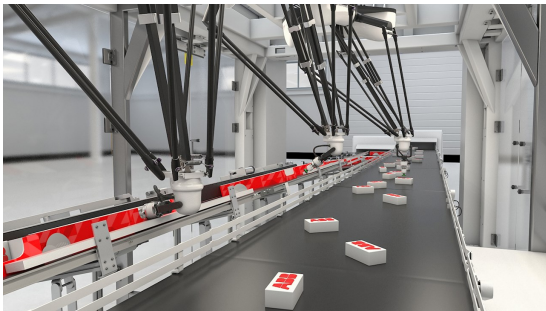
Historically, the development of industrial robotics has been driven by the needs of high-volume manufacturing sectors, such as the automotive and electronics industries. In these domains, robotic systems have become essential for modern production, with well-established applications in robotic welding, high-speed assembly (Fig. 1.3a), precision material handling (pick-and-place) 1.3c, and automated painting or coating 1.3d. These systems excel in structured and isolated environments where repeatability and high productivity are essential. Nowadays, robots cover many other market segments such as machinery, pharma, food and many others (Fig.1.4). Among the applications of industrial robots outside the factories, recently, interest has risen



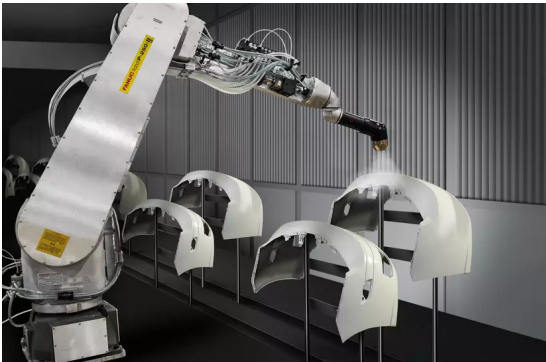
(a) Tesla assembly line (Source: Ryan Smith/Reddit).



(b) Ford line welding spot (Source: Oliver Berg/Getty Images)



(c) Five axes Delta robot for product picking (Source: ABB)



(d) Painting robots for hazardous environments (Source: FANUC)

Figure 1.3: Classical robot applications.

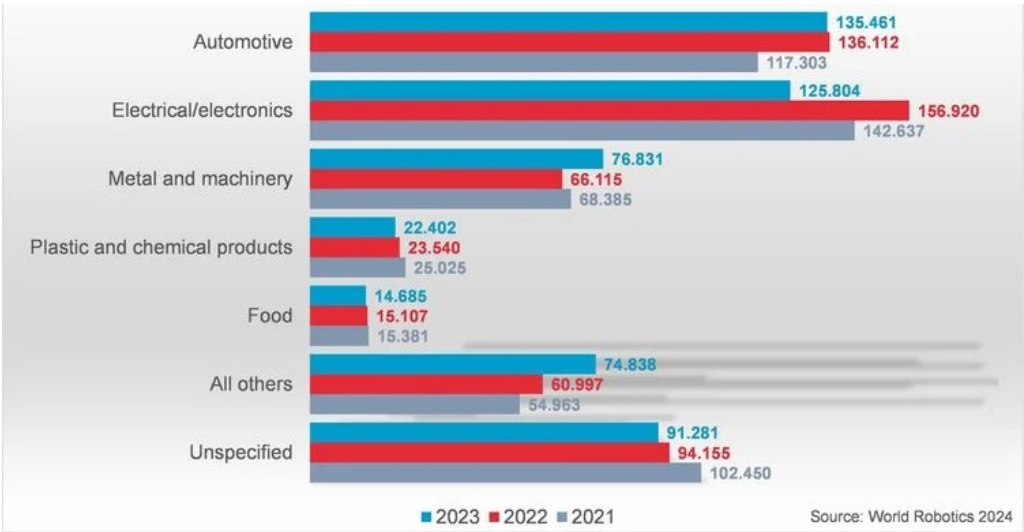


Figure 1.4: Annual installations of industrial robots by customer industry.



(a) TECLA Crane WASP, 3D printed housing (Source: WASP).



(b) MicroFactory for on-site home assembly (Source: AUAR).



(c) Automated laying of commercial blocks (Source: Fastbrick Robotics).



(d) Robotic construction painting (Source: PaintJet).

Figure 1.5: Robotic applications in building construction.

around the Architecture, Engineering, and Construction sector, characterised by more dynamic and less structured workspaces [6]. Emerging applications in this field now include large-scale 3D concrete printing [7] (Fig. 1.5a), automatic installation and assembly [8] (Fig. 1.5b), automatic bricklaying [9] (Fig. 1.5c), and automatic spraying [10] (Fig. 1.5d). Despite these advances, the automation of operations on the building envelope, such as maintenance, cleaning, and the assembly of complex facade components, presents a unique set of challenges due to the vast workspaces and verticality involved. This creates a critical need for robotic architectures that can provide both the necessary reach and the flexibility to adapt to diverse building geometries.

1.2 Cable-Driven Parallel Robots

Cable-Driven Parallel Robots (*CDPRs*) are a class of parallel manipulators in which the *EE*, integrated into a rigid moving platform, is controlled by flexible cables instead of rigid links [11] (Fig. 1.6). In the classical architecture, each cable is attached at one end to a servo-winch—typically mounted on an external rigid frame—and at the other end to the moving platform. Between these two points, the cable is routed through a transmission system composed of one or more pulleys. To ensure smooth guidance and reduce friction as the platform moves, the final pulley at the cable exit point is usually a *swiveling pulley*, designed to rotate around an axis tangent to its groove to accommodate the varying cable orientation [12]. The simultaneous winding and unwinding

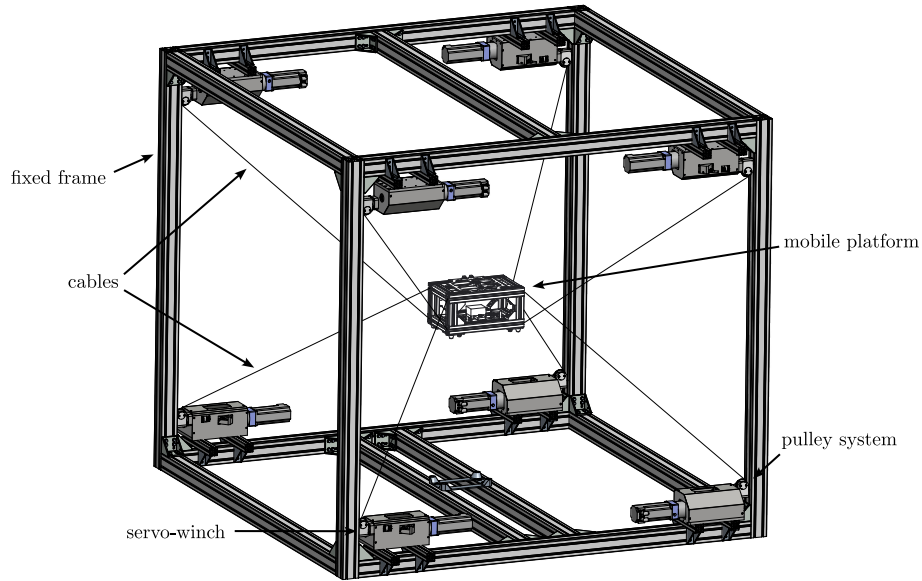


Figure 1.6: Model of a CDPR: the *moving platform* is connected to the *fixed frame* by *cables*, which are uncoiled by a set of *servo-winch*s and guided through *pulley systems*.

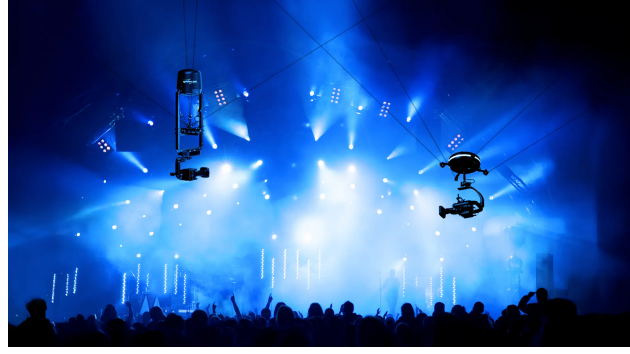
of the cables on the winch drums varies the cable lengths, allowing the *EE* to follow desired trajectories and perform manipulation tasks within the workspace.

Because cables are significantly lighter than traditional rigid links and are primarily subjected to tensile stresses, *CDPRs* exhibit an exceptionally high payload-to-weight ratio. This structural efficiency allows for the manipulation of heavy loads using a relatively lightweight and compact actuation system. Furthermore, the use of flexible cables enables unmatched scalability. Since the robot's mass does not increase proportionally with the size of its links, *CDPRs* can cover workspaces spanning hundreds of meters that would be physically or economically unfeasible for traditional parallel robots [13]. Another distinct advantage is their reconfigurability and deployability. The mechanical simplicity of the system allows cable exit points to be easily relocated to adapt the robot's workspace to specific task requirements or environmental constraints [14]. This adaptability makes *CDPRs* a versatile solution for unstructured and dynamic environments, such as construction sites, where the robot can be rapidly transported, installed, and modified as the project evolves.

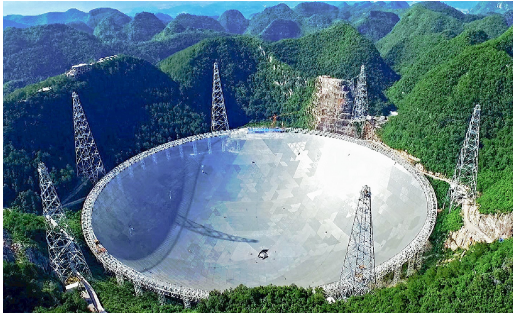
The unique structural characteristics of *CDPRs* make them particularly suitable for large-scale applications where traditional rigid-link manipulators are hindered by their own mass and workspace limitations. Historically, the first major development in this field is often considered to be the NIST RoboCrane, introduced in the early 1990s to manipulate heavy payloads for large-scale manufacturing, shipbuilding, and painting [15]. In the decades since, cable-suspended systems have seen significant commercial success in stadium event broadcasting through architectures like the SkyCam and Spidercam (Figs. 1.7a and 1.7b), which leverage the large workspaces attainable with cables to provide non-invasive aerial filming. On an even more monumental scale, the Five-hundred-meter Aperture Spherical Telescope (FAST) utilizes a large-scale *CDPR* to move a 30-ton receiver cabin with high precision across a vast focal plane (Fig. 1.7c). Beyond heavy lifting and vast reach, the inherent compliance of cables also enables specialized applications in the medical field, such as patient rehabilitation, where devices like the DIEGO system provide a safe and adjustable interface



(a) The SkyCam®(Source: SkyCam).



(b) Spidercam (Source: Ross).

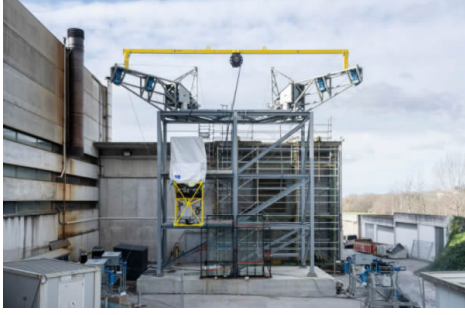
(c) The *FAST* telescope (Source: Hong Kong Laureate Forum).

(d) DIEGO device for Bilateral-Arm-Shoulder Therapy (Source: Tyromotion).

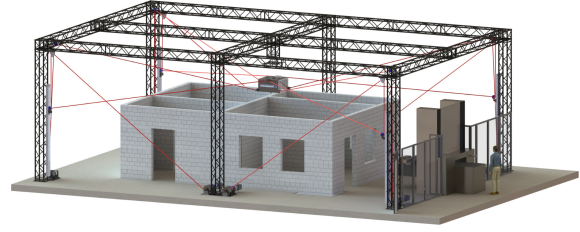
Figure 1.7: *CDPRs* applications

for limb therapy (Fig. 1.7d). Numerous proof-of-concept studies were proposed across different fields [16].

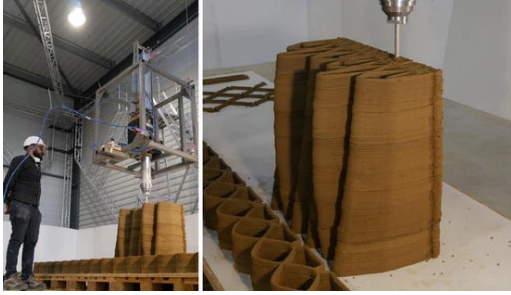
In recent years, research has increasingly focused on the Architecture, Engineering, and Construction (AEC) sector, an environment characterized by verticality and vast, unstructured workspaces that demand high operational flexibility. Notable advancements in this domain include the Hephaestus project [17,18], which developed a cable-driven solution for the automatic installation of curtain-wall modules on building façades (Fig. 1.8a), as well as various systems designed for robotic bricklaying [9,19–22] (Fig. 1.8b), large-scale 3D concrete printing [23–25] (Fig. 1.8c), and automated façade maintenance [26] (Fig. 1.8d). Because these construction environments are dynamic and temporary, the adaptability of *CDPRs* is further enhanced through deployable architectures that prioritize compactness and rapid installation. While some designs utilize lightweight portable structures such as masts or tripods [27,28], other approaches integrate the winches and transmission systems directly into the mobile platform to simplify transport [29,30]. However, this move toward high portability often necessitates the use of simplified winch designs that lack the complex spooling systems found in traditional industrial winches. While these simplified mechanisms reduce cost and weight, they introduce significant technical challenges, such as non-linear transmission ratios and uncertainties in the cable winding process that are difficult to model. As a result, indirect cable-length measurements obtained from motor encoders become less reliable, complicating the pose estimation and control of the robot. This creates a critical need for robust calibration and state-estimation strategies that can account



(a) The Hephaestus project (Credits: A. Iturralde).



(b) Robotic bricklaying (Source: [22]).



(c) CDPR large-scale 3D printing (Source: [24]).



(d) Kite Facade Inspection System (Source: Kite Robotics).

Figure 1.8: CDPRs applications in building construction and maintenance.

for the inherent uncertainties of deployable hardware, especially when the robot is frequently relocated or reconfigured in the field.

A common classification within the family of CDPRs is based on the state of constraint of the *EE*, which is formally defined by the relationship between the number of cables n and the *EE* DoFs n_d :

- **Underconstrained** ($n < n_d$), the number of cables is insufficient to fully constrain the *EE*. In this configuration, the *EE* cannot withstand an arbitrary external wrench, and at least one external action, such as gravity, is required to maintain cable tension. Furthermore, the *EE* retains passive degrees of freedom, meaning it can oscillate or move even when all actuators are locked;
- **Completely-constrained** ($n = n_d$), the *EE* configuration is fully determined by cable lengths, and the solution of the inverse dynamic (or static) problem allows a single set of cable tensions to be computed (under the rigid-body assumption);
- **Overconstrained** ($n > n_d$), the robot possesses more cables than the minimum required to fully constrain the *EE*. This structural redundancy makes the system statically indeterminate, meaning the inverse static problem admits infinite solutions for cable tensions. This redundancy is typically exploited to optimize internal force distribution, improve structural stiffness, or enlarge the reachable workspace.

It is worth noting that this classification is not intrinsically exhaustive, as the constraint condition is often an instantaneous property rather than a global one. Therefore, in

this context, n refers to the number of instantaneously taut cables rather than the total number of installed cables. For the sake of clarity throughout this thesis, we will refer to underconstrained architectures as Underactuated *CDPRs* (*UACDPRs*) and to overconstrained architectures as *OCDPRs*. This distinction is fundamental to isolating the unique control and calibration challenges associated with each architecture.

Despite the numerous advantages of *CDPRs*, their reliance on cables introduces significant theoretical and practical challenges. The most fundamental issue arises from the unilateral nature of cables, which can only exert tensile forces (pulling) and cannot support compression (pushing). To ensure that the robotic constraints remain active and the system remains controllable, cable tensions must be strictly maintained within a specific range: above a lower bound to prevent cable slackness and below an upper bound determined by the actuator's torque limits or the transmission breaking point. This results in a safety and control problem both for *OCDPRs* and *UACDPRs*. This requirement has led to the definition of the Wrench-Feasible Workspace (*WFW*), which is the set of end-effector poses where the robot can balance a given external wrench while keeping all cable tensions within these prescribed bounds [31–33]. Further details on the definition of the *WFW* are introduced in Sec 2.5.

In *OCDPRs*, the structural redundancy ($n > n_d$) leads to an infinite number of solutions for the inverse static or dynamic equilibrium, shifting the problem to finding an optimal Tension (or Force) Distribution (*TD*) [34–38]. Previous research has addressed this redundancy through various motion control techniques, such as Feedback Linearization [39], Sliding Mode Control [40], and Model Predictive Control (MPC) [41, 42]. A particularly effective and computationally efficient approach is Hybrid Position/Force Control in joint space (*HC*) [43, 44]. In this method, a number of cables n_d equal to the *EE* degrees of freedom are length-controlled to regulate the pose, while the remaining redundant cables are force-controlled to manage the internal tension distribution.

However, when employing *HC*, the specific selection of which cables are force-controlled and which are length-controlled is of primary importance, as it directly influences both the motion accuracy and the tension-distribution performance. Traditionally, force-controlled cables are chosen to optimize indices related solely to tension distribution [45]. However, these methods overlook how the actual motion control is realized and how sensitive it is to real-world perturbations. In practical applications, control strategies are highly sensitive to modeling inaccuracies and sensor noise [46]. For hybrid-controlled *OCDPRs*, this sensitivity can significantly shrink the usable workspace compared to the ideal *WFW* [47, 48], a challenge that motivates the study of error-insensitive workspace characterizations. For *UACDPRs*, the challenge is further compounded because the platform retains some degrees of freedom, meaning the workspace analysis must account not only for static equilibrium and tension constraints but also for stability at equilibrium [49].

A primary challenge in the practical use of *CDPRs* is the limited positioning accuracy resulting from the physical properties of the cables. Cables are difficult to model precisely because they are subject to creep, elongation under load, and, in large-scale applications, sagging due to their own mass. The cable length is crucial for state estimation and control of the *EE* [50], and can significantly affect the precision of the *CDPR* if errors are present in its estimate. The cable length is usually estimated indirectly from the encoder of the winch motor using a transmission model, which consists of a constant transmission ratio when classic winch design solutions are adopted [51].

While it may be more complex when deployable *CDPRs* are employed [52], in which a simplified winch design reduces space, cost, and weight, but it introduces uncertainties in the cable winding process and determines a nonlinear transmission ratio that depends on cable parameters that are difficult to estimate, such as friction and inertia [29]. As a result, indirect cable-length measurements for deployable *CDPRs* become less reliable [52].

Beyond these mechanical uncertainties, the performance of model-based control and state estimation is strictly bounded by the accuracy of the system's geometric and inertial parameters [53]. While a one-time factory calibration can be sufficient for rigid industrial robots, *CDPRs* often require frequent recalibration, especially with frequent reconfigurations. Every time a robot is relocated, or its cable exit points are adjusted to suit a new task, the geometric model must be updated to maintain operational accuracy [54, 55]. This requirement makes the use of external measurement systems, such as laser trackers or motion capture, impractical and cost-prohibitive for *CDPRs*. Consequently, research has shifted toward self-calibration, a procedure that utilizes only the robot's onboard sensors to identify its model parameters [55–59]. A particularly critical subset of this field is initial-pose self-calibration. Most *CDPRs* rely on relative internal sensors, such as incremental rotary encoders, which only measure variations from an unknown starting point [52, 56, 57, 59–61]. At every system startup, the absolute reference values of these sensors, and by extension, the initial pose of the *EE*, are unknown.

The initial-pose self-calibration can be formulated as an overdetermined nonlinear optimization problem. It is typically formulated using Nonlinear Weighted Least Squares (*NWLS*), where the objective is to minimize the *residuals*, namely the difference between actual sensor readings and the values predicted by the robot's kinematic or static models. This problem presents a practical difficulty: to find the calibration parameters, the robot must be moved to different measurement poses. However, because the initial state and model are unknown or imprecise, these poses themselves are unknown and must be estimated as part of the optimization. If this initial state is estimated incorrectly, it can lead to cable slack, collisions, or a total loss of controllability during the robot's first movements.

The success of such a procedure depends on three main pillars:

- **the sensor set**, while minimal equipment consists of winch encoders for cable length variations, accuracy can be significantly improved by incorporating alternative sensors. These include incremental encoders on swivel pulleys to measure exit angles, load cells for cable tensions, and Attitude and Heading Reference Systems (AHRS), or inclinometers, to provide direct *EE* orientation data;
- **the measurement pose selection**, the number and distribution of measurement poses must ensure the observability of the system. This requires a set of configurations that "excite" all the variables to be estimated, resulting in a well-conditioned sensitivity or Jacobian matrix.
- **the numerical solvers**, since the optimization is highly non-linear, the choice of solver, such as Levenberg-Marquardt or Trust-Region Newton methods, and the quality of the initial guess are fundamental to ensuring global convergence.

For *UACDPRs*, these challenges are even more significant. Since the *EE* retains some degrees of freedom and may oscillate like a pendulum, data must often be col-

lected in static postures after oscillations have subsided, or through manual intervention by an operator. Despite the theoretical and technical solutions proposed in the literature, past state-of-the-art methods required several minutes for automatic data acquisition [62]. This demonstrated the need for rapid, fully automated, and robust self-calibration procedures that can operate effectively under the high sensor uncertainties typical of large-scale and deployable systems.

1.3 Thesis Goals and Outline

The primary objective of this thesis is to enhance the operational accuracy and reliability of Cable-Driven Parallel Robots. This objective is pursued through two main research directions: a control-oriented workspace analysis for overconstrained architectures and a comprehensive investigation of initial-pose self-calibration techniques for underactuated, overconstrained, and deployable systems.

The first research line focuses on *OCDPRs* employing Hybrid Position/Force Control (*HC*) in joint space. While *HC* offers a computationally efficient way to manage redundancy, its performance is highly dependent on the choice of force-controlled cables and is sensitive to modeling inaccuracies. To address this, this thesis introduces the concept of the Wrench-Feasible Error-Insensitive Workspace (*WFEIW*). The *WFEIW* provides a theoretical framework for evaluating accuracy and control-error sensitivity, identifying the subset of poses where user-defined performance is guaranteed despite bounded sensor noise and control uncertainties. In the *WFEIW*, user-defined accuracy requirements and cable tensions within predetermined bounds are guaranteed, while accounting for sensor-error expectations. Based on this analysis, a nested optimization algorithm is developed to maximize the *WFEIW* volume by determining: (i) the most suitable combination of length- and force-controlled cables, and (ii) the optimal platform geometry. Simulations are conducted on well-known *OCDPR* architectures as application examples of the control-strategy and design optimization algorithm.

Initial-pose self-calibration techniques are then investigated thoroughly. Regarding underactuated architectures, the research introduces rapid and fully automated data-acquisition algorithms for the initial-pose self-calibration of *UACDPRs*. These procedures significantly reduce the overall calibration time. Two novel acquisition strategies are proposed: a static-equilibrium method that, along with measurements of cable length variations, swivel angle variations, and *AHRS* orientation, incorporates cable-tension data; and a faster method based solely on geometric constraints. By formulating the problem as a Nonlinear Weighted Least-Squares (*NWLS*) optimization and implementing a systematic heuristic for initial-guess generation, these methods enable high-accuracy pose estimation while enhancing the degree of automation during system startup. Then, a systematic methodology and unified framework are introduced to evaluate the influence of key setup parameters on self-calibration performance. Through an extensive Monte Carlo simulation campaign, and an experimental validation covering both overconstrained and underactuated robots, the study analyzes the impact of the number and distribution of measurement poses, the choice of onboard sensor sets, and the robustness of various nonlinear solvers. This analysis utilizes genetic algorithms to maximize observability indices for pose selection and compares iterative solvers—including the Levenberg-Marquardt algorithm and Trust-Region Newton methods using real Hessian information—to provide practical guide-

lines for designing efficient calibration procedures for any *CDPR* architecture. Finally, the work investigates the application of initial-pose self-calibration to deployable *CDPRs*, analyzing how the choice of sensors affects accuracy when simplified transmissions introduce nonlinearity and uncertainty. To further streamline the initialization phase, novel online self-calibration techniques based on Recursive Least Squares (*RLS*) and Kalman filters are proposed as a faster alternative to traditional sequential online approaches [58,59], aiming to maintain estimation accuracy while further reducing the total time required for the robot to become operational.

The manuscript is organized as follows.

Chapter 2 introduces the kinematic and static models of both classic and deployable *CDPRs*, establishing the mathematical foundations for workspace computation and tension distribution.

Chapter 3 presents the control-based workspace analysis, detailing the mathematical formulation of the *WFEIW* and the nested optimization algorithm for *OCDPR* design.

Chapter 4 provides a comprehensive analysis of initial-pose self-calibration methods, encompassing automated data acquisition, sensor set evaluation, and the systematic study of setup parameters.

Finally, Chapter 5 summarizes the findings and outlines future research directions to improve *CDPR* autonomy and accuracy.

Chapter 2

Modeling

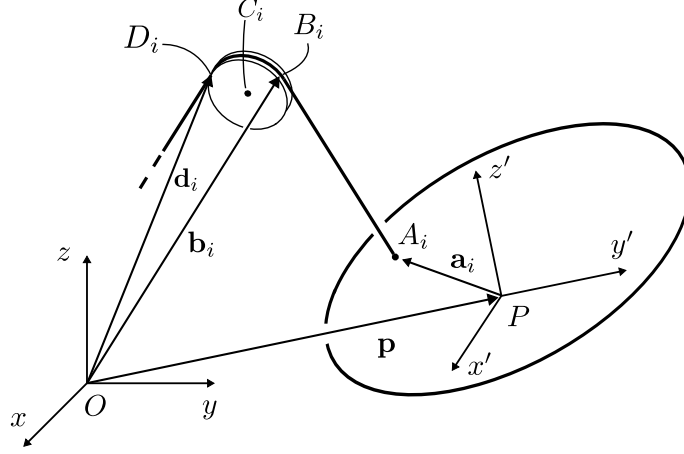
In this chapter, we establish the mathematical foundations for generic *CDPRs* with n cables and $n_d = 6$ degrees of freedom (*DoFs*) moving in $SE(3)$ ¹. Following the assumptions commonly adopted for the scale of manipulators considered in this research, cables are assumed to be massless and inextensible. Consequently, complex phenomena such as sagging and elongation are neglected, and cables are modeled as straight line segments coiled and uncoiled by servo-controlled winches. Nominally, cable lengths are assumed to vary proportionally to actuator displacements, which is equivalent to assuming a constant transmission ratio at the winch level.

Furthermore, the geometry of the swivel pulleys used to route the cables inside the workspace is explicitly considered to ensure high modeling accuracy. In practice, the variable section of each cable is defined as the segment between the attachment point on the *EE*, typically realized with a swivel hoist ring, and the cable entry point on the swiveling pulley. To account for diverse hardware configurations and deployment scenarios, we distinguish between two primary architectures:

- **Classical architecture:** Detailed in Sections 2.1, 2.1.1, 2.1.2, and 2.2, this configuration features winches and pulley systems mounted on an external fixed frame.
- **Deployable architecture:** Described in Sections 2.3, 2.3.1, 2.3.2, and 2.4, this configuration integrates both winches and pulley transmissions directly into the moving platform to enhance portability and ease of installation.

To maintain a consistent and lean notation throughout these models, the prime symbol $(\cdot)'$ is used to denote quantities expressed in the mobile frame $Px'y'z'$. Sections 2.1 and 2.3 present the geometric modeling for each architecture, followed by the first-order differential kinematics in Sections 2.1.1 and 2.3.1, and the second-order derivations in Sections 2.1.2 and 2.3.2. The static models are then established in Sections 2.2 and 2.4. Finally, Section 2.5 introduces the fundamental definitions for workspace analysis and tension distribution, providing the theoretical basis for the chapters that follow.

¹The content of this chapter is partially based on [48, 61] submitted to *Mechanism and Machine Theory* and *Robotica*.

Figure 2.1: Geometric model of the i -th cable transmission.

2.1 Geometric model

We consider a spatial *CDPR* where the *EE* has six degrees of freedom ($n_d = 6$) and is actuated by n cables. To describe the system, we define an inertial frame $Oxyz$ and a moving frame $Px'y'z'$ attached to the *EE*. The pose of the *EE* is defined by the position vector $\mathbf{p}$ of the origin P and by the rotation matrix $\mathbf{R}(\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon} = [\phi \ \theta \ \chi]^T$ is a minimal orientation parametrization. Specifically, we adopt *XYZ* Euler angles, where the rotation matrix is computed as:

$$\mathbf{R}(\boldsymbol{\epsilon}) = \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\chi), \quad (2.1)$$

with $R_x(\cdot)$, $R_y(\cdot)$, and $R_z(\cdot)$ representing elementary rotations about the x , y , and z axes, respectively. The generalized coordinate vector of the system is thus $\boldsymbol{\zeta} = [\mathbf{p}^T \ \boldsymbol{\epsilon}^T]^T \in \mathbb{R}^6$. Each cable is routed from a winch fixed on the frame to the *EE* through a swiveling pulley transmission (Fig. 2.1). For the i -th cable, the cable enters the pulley groove at a fixed entry point D_i , exits at a variable point B_i , and is attached to the *EE* at point A_i . Following the approach introduced in [63], the variable position of B_i in the inertial frame is explicitly modeled to account for the pulley's influence on the effective cable length.

For each pulley, we introduce a local reference frame $D_i x_i y_i z_i$ with constant unit vectors $\{\mathbf{i}_i, \mathbf{j}_i, \mathbf{k}_i\}$ (Figs. 2.2a and 2.2b). The pulley, which has a radius r_i and center C_i , is tangent to the local z_i axis at the entry point D_i and swivels about this axis by an angle σ_i . Let $\mathbf{u}_i$ be the unit vector pointing from D_i to C_i , and let $\mathbf{w}_i$ be the unit vector orthogonal to the pulley groove plane, obtained as $\mathbf{w}_i = \mathbf{k}_i \times \mathbf{u}_i$ (where the symbol $\times$ denotes the cross product between two vectors). The unit vector $\mathbf{n}_i$, parametrized by the tangency angle ψ_i , points from C_i to B_i , while the direction of the rectilinear cable segment is given by $\mathbf{t}_i = \mathbf{w}_i \times \mathbf{n}_i$. These vectors satisfy the following relationships:

$$\mathbf{u}_i = \cos \sigma_i \mathbf{i}_i + \sin \sigma_i \mathbf{j}_i, \quad (2.2)$$

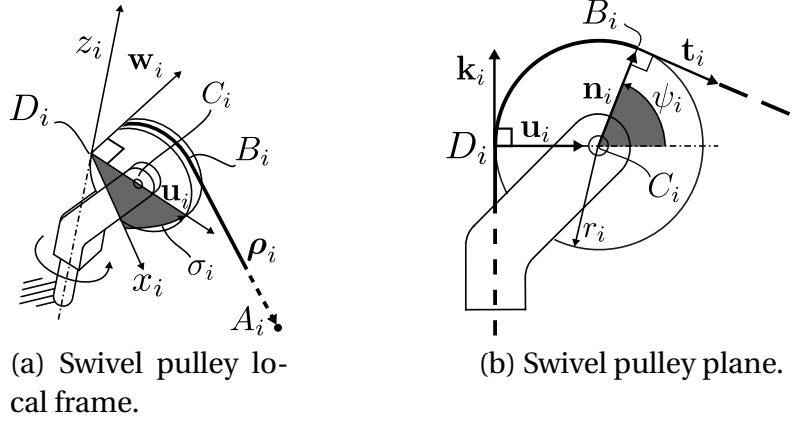
$$\mathbf{w}_i = -\sin \sigma_i \mathbf{i}_i + \cos \sigma_i \mathbf{j}_i, \quad (2.3)$$

$$\mathbf{n}_i = \cos \psi_i \mathbf{u}_i + \sin \psi_i \mathbf{k}_i, \quad (2.4)$$

$$\mathbf{t}_i = \sin \psi_i \mathbf{u}_i - \cos \psi_i \mathbf{k}_i. \quad (2.5)$$

Let the constant vector $\mathbf{d}_i$ denote the position of point D_i in the inertial frame. The position of the exit point B_i follows from the pulley geometry as:

$$\mathbf{b}_i = \mathbf{d}_i + r_i(\mathbf{u}_i + \mathbf{n}_i). \quad (2.6)$$

Figure 2.2: i -th swivel pulley geometric model

Similarly, let $\mathbf{a}'_i$ be the constant coordinates of the attachment point A_i in the local EE frame. In the inertial frame, the position vector pointing from P to A_i is given by:

$$\mathbf{a}_i = \mathbf{R}\mathbf{a}'_i. \quad (2.7)$$

We define the vector between the pulley entry point D_i and the platform attachment point A_i as $\boldsymbol{\rho}_i = \mathbf{p} + \mathbf{a}_i - \mathbf{d}_i$. Assuming the cable remains coplanar with the pulley groove, the swivel angle σ_i is determined by the following constraint:

$$\mathbf{w}_i \cdot \boldsymbol{\rho}_i = 0. \quad (2.8)$$

Substituting Eq. (2.3) into Eq. (2.8) yields the i -th swivel angle

$$\sigma_i = \text{atan2}(\mathbf{j}_i \cdot \boldsymbol{\rho}_i, \mathbf{i}_i \cdot \boldsymbol{\rho}_i). \quad (2.9)$$

A detailed derivation of this result is provided in Appendix A.1.

Let the cable vector be $\boldsymbol{\rho}_i = \mathbf{p} + \mathbf{a}_i - \mathbf{b}_i$. Substituting Eq. (2.6) in the latter, we obtain:

$$\boldsymbol{\rho}_i = \boldsymbol{\rho}_i - r_i(\mathbf{u}_i + \mathbf{n}_i) = \|\boldsymbol{\rho}_i\| \mathbf{t}_i. \quad (2.10)$$

The tangency condition at B_i requires the cable vector $\boldsymbol{\rho}_i$ to be orthogonal to the radial vector $\mathbf{n}_i$:

$$\mathbf{n}_i \cdot \boldsymbol{\rho}_i = 0, \quad (2.11)$$

This condition, allows to compute the tangency angle ψ_i (see App. A.2) as:

$$\psi_i = 2 \arctan \left(\frac{\varrho_{k_i}}{\varrho_{u_i}} + \sqrt{\left(\frac{\varrho_{k_i}}{\varrho_{u_i}} \right)^2 + 1 - \frac{2r_i}{\varrho_{u_i}}} \right), \quad (2.12)$$

where $\varrho_{k_i} = \boldsymbol{\rho}_i \cdot \mathbf{k}_i$ and $\varrho_{u_i} = \boldsymbol{\rho}_i \cdot \mathbf{u}_i$.

Finally, the total cable length l_i is defined by the sum of the rectilinear segment $\|\boldsymbol{\rho}_i\|$ and the arc $\widehat{D_i B_i}$ wrapped around the pulley:

$$l_i = \|\boldsymbol{\rho}_i\| + r_i(\pi - \psi_i). \quad (2.13)$$

The variables l_i , ψ_i , and σ_i (for $i = 1, \dots, n$) represent the joint coordinates of the CDPR, where the cable lengths are the active (actuated) joints and the tangency and

swivel angles are the passive joints. Consequently, Eqs. (2.9), (2.12) and (2.13) constitute the inverse kinematic (*IK*) model of the robot, enabling the determination of the joint variables for a given end-effector pose ζ . For any specified configuration, the *IK* problem is decoupled for each kinematic chain and always admits a unique physical solution, a property that holds for both *OCDPRs* and *UACDPRs*.

It is important to emphasize, however, that the existence of a geometric *IK* solution does not imply physical feasibility. These purely kinematic relationships do not account for whether a pose ζ is actually reachable, dynamically achievable, or, particularly for *UACDPRs*, constitutes a stable equilibrium pose. Determining the set of poses the *EE* can effectively attain, referred to as *workspace analysis*, requires the solution of the robot's statics or dynamics to ensure that all cable tensions remain within the prescribed bounds. This analysis is detailed in Section 2.5.

2.1.1 First-order kinematics

Let $\omega \in \mathbb{R}^3$ denote the angular velocity of the *EE*. For a minimal orientation parametrization $\epsilon = [\phi \ \theta \ \chi]^T$, the relationship between the orientation rates $\dot{\epsilon}$ and the angular velocity ω is expressed as:

$$\omega = \mathbf{H}(\epsilon) \dot{\epsilon}, \quad (2.14)$$

where $\mathbf{H}(\epsilon) \in \mathbb{R}^{3 \times 3}$ depends on the chosen parametrization [2]. For the *XYZ* Euler angles parameterization introduced in Eq. 2.1, $\mathbf{H}$ is given by

$$\mathbf{H}(\epsilon) = \begin{bmatrix} 1 & 0 & \sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & \sin \phi & \cos \phi \cos \theta \end{bmatrix}. \quad (2.15)$$

Defining the *EE* twist as $\mathbf{v} = [\dot{\mathbf{p}}^T \ \omega^T]^T \in \mathbb{R}^6$, the transformation between the twist and the generalized coordinate derivative $\dot{\zeta} = [\dot{\mathbf{p}}^T \ \dot{\epsilon}^T]^T$ is:

$$\mathbf{v} = \mathbf{D}(\epsilon) \dot{\zeta}, \quad \mathbf{D}(\epsilon) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{H}(\epsilon) \end{bmatrix}, \quad (2.16)$$

with $\mathbf{I}_{3 \times 3}$ and $\mathbf{0}_{3 \times 3}$ the 3×3 identity and zero matrices, respectively.

The rate of change of the i -th swivel angle, $\dot{\sigma}_i$, is obtained by differentiating the pulley coplanarity constraint (Eq. 2.8) with respect to time. As derived in Appendix A.3, this relates to the twist through the vector ξ_{σ_i} :

$$\dot{\sigma}_i = \xi_{\sigma_i}^T \mathbf{v}, \quad \xi_{\sigma_i} = \frac{1}{\rho_{u_i}} \begin{bmatrix} \mathbf{w}_i \\ \mathbf{a}_i \times \mathbf{w}_i \end{bmatrix}. \quad (2.17)$$

Similarly, the rate of change of the i -th cable length, $\dot{l}_i$, is found by differentiating the geometric identity $\|\rho_i\|^2 = \rho_i^T \rho_i$. As detailed in Appendix A.5, the resulting mapping is:

$$\dot{l}_i = \xi_{l_i}^T \mathbf{v}, \quad \xi_{l_i} = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{a}_i \times \mathbf{t}_i \end{bmatrix}. \quad (2.18)$$

Following a similar approach, the rate of change for the i -th tangency angle, $\dot{\psi}_i$, may also be determined. While its explicit derivation is provided in Appendix A.4, as it is needed for the second-order analysis in Section 2.1.2, its formulation is omitted here

to maintain focus on the primary joint-space variables required for the subsequent discussions.

By stacking these relations for all n cables, we define the geometric Jacobians Ξ_σ and $\Xi_l \in \mathbb{R}^{n \times 6}$:

$$\dot{\boldsymbol{\sigma}} = \Xi_\sigma \mathbf{v}, \quad \Xi_\sigma = [\boldsymbol{\xi}_{\sigma_1} \quad \dots \quad \boldsymbol{\xi}_{\sigma_n}]^T, \quad (2.19)$$

$$\dot{\mathbf{l}} = \Xi_l \mathbf{v}, \quad \Xi_l = [\boldsymbol{\xi}_{l_1} \quad \dots \quad \boldsymbol{\xi}_{l_n}]^T, \quad (2.20)$$

To obtain the analytical Jacobians, $\mathbf{J}_\sigma$ and $\mathbf{J}_l$, which map the joint rates directly to the generalized coordinate rates $\dot{\boldsymbol{\zeta}}$, we substitute Eq. (2.16) into the geometric mappings:

$$\dot{\boldsymbol{\sigma}} = \mathbf{J}_\sigma \dot{\boldsymbol{\zeta}}, \quad \mathbf{J}_\sigma = \Xi_\sigma \mathbf{D}, \quad (2.21)$$

$$\dot{\mathbf{l}} = \mathbf{J}_l \dot{\boldsymbol{\zeta}}, \quad \mathbf{J}_l = \Xi_l \mathbf{D}, \quad (2.22)$$

The rows of these analytical Jacobians correspond to the partial derivatives of the joint variables with respect to the EE pose:

$$\frac{\partial \sigma_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{D}(\boldsymbol{\epsilon}), \quad (2.23)$$

$$\frac{\partial l_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{l_i}^T \mathbf{D}(\boldsymbol{\epsilon}). \quad (2.24)$$

2.1.2 Second-order kinematics

The workspace analysis in Chapter 3 and the non-linear self-calibration algorithms in Chapter 4 require second-order kinematic information. Consequently, the Hessians of the swivel angles $\sigma_i(\boldsymbol{\zeta})$ and cable lengths $l_i(\boldsymbol{\zeta})$ are derived here.

From the analytical Jacobians defined in Eqs. (2.21) and (2.22), the partial derivatives of $\mathbf{D}$, $\boldsymbol{\xi}_{\sigma_i}$, and $\boldsymbol{\xi}_{l_i}$ with respect to the pose $\boldsymbol{\zeta}$ are required. As established in Eq. (2.16), the matrix $\mathbf{D}(\boldsymbol{\epsilon})$ depends on the pose only through the orientation parameters $\boldsymbol{\epsilon}$. Thus, its derivatives with respect to the position components $\mathbf{p}$ are zero, while its derivatives with respect to the Euler angles ϕ , θ , and χ are:

$$\frac{\partial \mathbf{D}}{\partial \phi} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{\partial \mathbf{H}}{\partial \phi} \end{bmatrix}, \quad \frac{\partial \mathbf{D}}{\partial \theta} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{\partial \mathbf{H}}{\partial \theta} \end{bmatrix}, \quad \frac{\partial \mathbf{D}}{\partial \chi} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{\partial \mathbf{H}}{\partial \chi} \end{bmatrix}. \quad (2.25)$$

For the XYZ Euler parametrization, the partial derivatives of $\mathbf{H}$ with respect to ϕ , θ , and χ are:

$$\frac{\partial \mathbf{H}}{\partial \phi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin \phi & -\cos \phi \cos \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \end{bmatrix}, \quad \frac{\partial \mathbf{H}}{\partial \theta} = \begin{bmatrix} 0 & 0 & \cos \theta \\ 0 & 0 & \sin \phi \sin \theta \\ 0 & 0 & -\cos \phi \sin \theta \end{bmatrix}, \quad \frac{\partial \mathbf{H}}{\partial \chi} = \mathbf{0}_{3 \times 3}. \quad (2.26)$$

Using the chain rule on the first-order analytical Jacobian (2.23), the Hessian of the i -th swivel angle, $\partial^2 \sigma_i / \partial \boldsymbol{\zeta}^2 \in \mathbb{R}^{6 \times 6}$, is:

$$\frac{\partial^2 \sigma_i}{\partial \boldsymbol{\zeta}^2} = \frac{\partial \boldsymbol{\xi}_{\sigma_i}^T}{\partial \boldsymbol{\zeta}} \mathbf{D}(\boldsymbol{\epsilon}) + \sum_{j=1}^6 \xi_{\sigma_i, j} \frac{\partial \mathbf{D}_{(j,:)}}{\partial \boldsymbol{\zeta}}, \quad (2.27)$$

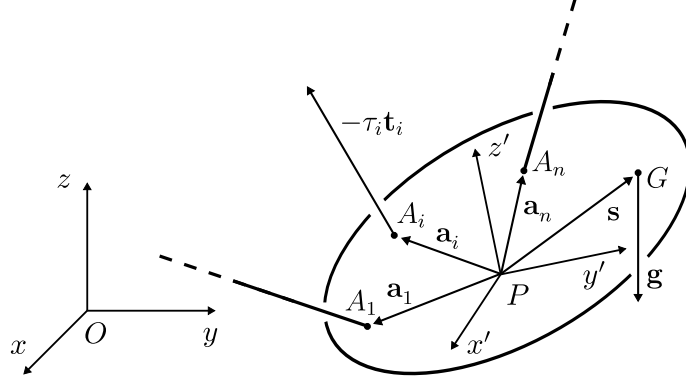


Figure 2.3: EE Free-Body Diagram

where $\xi_{\sigma_i,j}$ denotes the j -th component of ξ_{σ_i} and $\mathbf{D}_{(j,:)}$ denotes the j -th row of $\mathbf{D}$. The derivation of $\partial \xi_{\sigma_i}^T / \partial \zeta$ is detailed in A.6 and the term can be expressed as

$$\frac{\partial \xi_{\sigma_i}^T}{\partial \zeta} = \mathbf{D}^T \bar{\xi}_{\sigma_i}, \quad \bar{\xi}_{\sigma_i} = \frac{1}{\rho_{u_i}} (-\xi_{u_i} \xi_{\sigma_i}^T - \xi_{\sigma_i} \xi_{u_i}^T + \mathbf{A}_{w_i}), \quad (2.28)$$

where the auxiliary matrix $\mathbf{A}_{w_i}$ and the vector ξ_{u_i} are defined as:

$$\mathbf{A}_{w_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{a}}_i \tilde{\mathbf{w}}_i \end{bmatrix}, \quad \xi_{u_i} = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{a}_i \times \mathbf{u}_i \end{bmatrix}. \quad (2.29)$$

Here, the symbol $\tilde{(\cdot)}$ denotes the skew-symmetric matrix used to represent the cross-product operation.

Similarly, the Hessian of the i -th cable length, $\partial^2 l_i / \partial \zeta^2 \in \mathbb{R}^{6 \times 6}$, is computed as:

$$\frac{\partial^2 l_i}{\partial \zeta^2} = \frac{\partial \xi_{l_i}^T}{\partial \zeta} \mathbf{D}(\epsilon) + \sum_{j=1}^6 \xi_{l_i,j} \frac{\partial \mathbf{D}_{(j,:)}}{\partial \zeta}, \quad (2.30)$$

where $\xi_{l_i,j}$ denotes the j -th component of ξ_{l_i} . The derivation $\partial \xi_{l_i}^T / \partial \zeta$ is detailed in A.7 and the term can be written as

$$\frac{\partial \xi_{l_i}^T}{\partial \zeta} = \mathbf{D}^T \bar{\xi}_{l_i}, \quad \bar{\xi}_{l_i} = \left(\rho_{u_i} \sin(\psi_i) \xi_{\sigma_i} \xi_{\sigma_i}^T + \|\rho_i\| \xi_{\psi_i} \xi_{\psi_i}^T + \mathbf{A}_{t_i} \right), \quad (2.31)$$

with the orientation-dependent terms defined as:

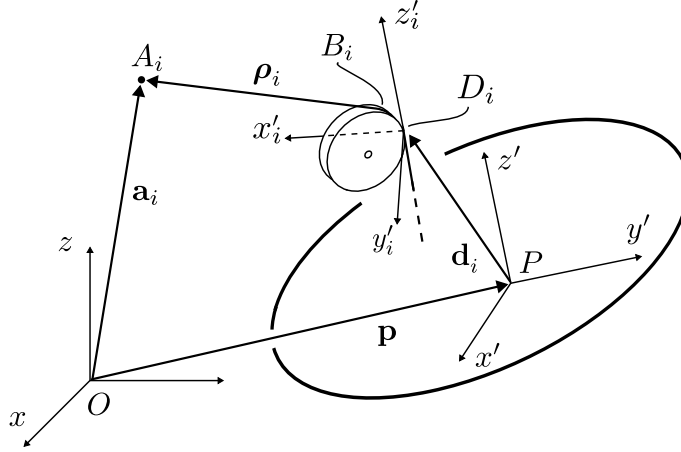
$$\mathbf{A}_{t_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{a}}_i' \tilde{\mathbf{t}}_i \end{bmatrix}, \quad \xi_{\psi_i} = \frac{1}{\|\rho_i\|} \begin{bmatrix} \mathbf{n}_i \\ \mathbf{a}_i' \times \mathbf{n}_i \end{bmatrix}. \quad (2.32)$$

2.2 Static model

The *CDPR* static model is obtained from the mechanical equilibrium of the *EE* (Fig. 2.3). The model is formulated under the assumption that all cables remain taut and that gravity is the only external force acting on the *EE*, so that:

$$\Xi_l^T \boldsymbol{\tau} = \mathbf{f}, \quad \text{with} \quad \mathbf{f} = \begin{bmatrix} \mathbf{g} \\ \mathbf{s} \times \mathbf{g} \end{bmatrix}, \quad (2.33)$$

where $\mathbf{g} = [0, 0, -mg]^T$, m is the *EE* mass, g is the gravitational acceleration, $\mathbf{s}$ is the position vector from P to the center of mass G , and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the array of cable tensions. $\Xi_l^T \in \mathbb{R}^{6 \times n}$ is the geometric Jacobian transposed (also often called the structure matrix).

Figure 2.4: Geometric model of the i -th cable transmission for deployable *CDPRs*.

2.3 Geometric model of deployable *CDPRs*

For deployable architectures, we consider a spatial *CDPR* with $n_d = 6$ degrees of freedom and n cables. The fundamental modeling assumptions remain unchanged: cables are treated as straight, massless, and inextensible lines. We retain the inertial frame $Oxyz$ and the mobile frame $Px'y'z'$ attached to the platform, as well as the pose vector ζ and the rotation matrix $\mathbf{R}(\epsilon)$. Consistent with Section 2.1, the orientation $\epsilon = [\phi \ \theta \ \chi]^T$ follows the *XYZ* Euler Angles convention.

The distinguishing feature of this architecture is the integration of the winches and pulley transmissions directly into the mobile platform. Consequently, the transmission geometry is fixed relative to the platform rather than the frame. As in the classical case, each cable is routed through a swiveling pulley (Fig. 2.4). In the following, quantities expressed in the mobile frame $Px'y'z'$ are denoted with a prime $(\cdot)'$.

For the i -th cable, the entry point D_i and the exit point B_i are part of the platform assembly, while the anchor point A_i is fixed on the external frame. To describe the pulley geometry, we introduce a local frame $D_ix'_iy'_iz'_i$ with unit vectors $\{\mathbf{i}'_i, \mathbf{j}'_i, \mathbf{k}'_i\}$ that are constant in the mobile frame $Px'y'z'$ (Figs. 2.5a and 2.5b). Each pulley of radius r_i and center C_i is tangent to the local axis z'_i at D_i and swivels about z'_i by the angle σ_i . If the unit vector $\mathbf{u}'_i$ points from D_i to C_i , and the unit vector $\mathbf{w}'_i$ is orthogonal to the pulley groove plane, then $\mathbf{w}'_i = \mathbf{k}'_i \times \mathbf{u}'_i$. The tangency angle ψ_i defines the orientation of the unit vector $\mathbf{n}'_i$, which points from C_i to B_i . The unit vector $\mathbf{t}'_i$ along the i -th cable is given by relation $\mathbf{t}'_i = \mathbf{w}'_i \times \mathbf{n}'_i$. Accordingly, the unit vectors can be described as

$$\mathbf{u}'_i = \cos \sigma_i \mathbf{i}'_i + \sin \sigma_i \mathbf{j}'_i, \quad (2.34)$$

$$\mathbf{w}'_i = -\sin \sigma_i \mathbf{i}'_i + \cos \sigma_i \mathbf{j}'_i, \quad (2.35)$$

$$\mathbf{n}'_i = \cos \psi_i \mathbf{u}'_i + \sin \psi_i \mathbf{k}'_i, \quad (2.36)$$

$$\mathbf{t}'_i = \sin \psi_i \mathbf{u}'_i - \cos \psi_i \mathbf{k}'_i, \quad (2.37)$$

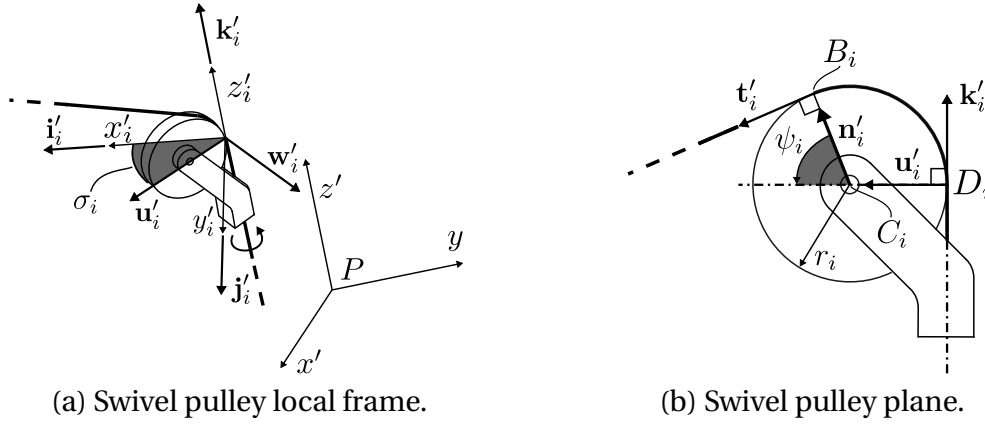
in the mobile frame, and as

$$\mathbf{u}_i = \mathbf{R} \mathbf{u}'_i, \quad (2.38)$$

$$\mathbf{w}_i = \mathbf{R} \mathbf{w}'_i, \quad (2.39)$$

$$\mathbf{n}_i = \mathbf{R} \mathbf{n}'_i, \quad (2.40)$$

$$\mathbf{t}_i = \mathbf{R} \mathbf{t}'_i \quad (2.41)$$


 Figure 2.5: i -th swivel pulley geometric model

in the inertial frame.

The relative position between P and D_i is given by $\mathbf{d}_i = \mathbf{R}\mathbf{d}'_i$, where $\mathbf{d}'_i$ is constant in the mobile frame. The position of A_i in the inertial frame is given by $\mathbf{a}_i$. Defining the vector $\boldsymbol{\varrho}_i = A_i - D_i$, we have:

$$\boldsymbol{\varrho}_i = \mathbf{a}_i - \mathbf{p} - \mathbf{d}_i. \quad (2.42)$$

Assuming that the cable remains coplanar with the swivel pulley, the resulting constraint is:

$$\mathbf{w}_i \cdot \boldsymbol{\varrho}_i = 0 \quad (2.43)$$

From the latter, the swivel angle is found through inverse kinematics as:

$$\sigma_i = \text{atan2}(\mathbf{j}_i \cdot \boldsymbol{\varrho}_i, \mathbf{i}_i \cdot \boldsymbol{\varrho}_i). \quad (2.44)$$

Considering the pulley kinematics, the cable vector $\boldsymbol{\rho}_i = A_i - B_i$ in the inertial frame is:

$$\boldsymbol{\rho}_i = \boldsymbol{\varrho}_i - r_i(\mathbf{u}_i + \mathbf{n}_i). \quad (2.45)$$

Since the cable is tangent to the pulley groove in B_i , the resulting geometrical constraint is:

$$\mathbf{n}_i \cdot \boldsymbol{\rho}_i = 0. \quad (2.46)$$

Thus, the tangency angle can be computed as:

$$\psi_i = 2 \arctan \left(\frac{\varrho_{k_i}}{\varrho_{u_i}} + \sqrt{\left(\frac{\varrho_{k_i}}{\varrho_{u_i}} \right)^2 + 1 - \frac{2r_i}{\varrho_{u_i}}} \right). \quad (2.47)$$

where $\varrho_{k_i} = \boldsymbol{\varrho}_i \cdot \mathbf{k}_i$ and $\varrho_{u_i} = \boldsymbol{\varrho}_i \cdot \mathbf{u}_i$. Finally, the total cable length l_i is determined by the sum of the rectilinear segment and the pulley arc:

$$l_i = \|\boldsymbol{\rho}_i\| + r_i(\pi - \psi_i). \quad (2.48)$$

It is important to note that the derivations for σ_i and ψ_i are mathematically equivalent to those presented for the classical architecture, and thus follow the proofs provided in Appendices A.1 and A.2.

2.3.1 First-order kinematics of deployable *CDPRs*

The first-order differential kinematics for deployable architectures follow a similar logic to the classical case, mapping the *EE* velocity to the rates of change of the joint variables. The relationship between the *EE* twist $\mathbf{v}$ and the generalized coordinate derivative $\dot{\boldsymbol{\zeta}}$ is identical to that defined in Section 2.1.1:

$$\mathbf{v} = \mathbf{D}(\boldsymbol{\epsilon})\dot{\boldsymbol{\zeta}}, \quad (2.49)$$

where $\mathbf{D}(\boldsymbol{\epsilon})$ remains the transformation matrix defined in Eq. (2.16), depending on the *XYZ* Euler angle parametrization.

Following the derivations detailed in Appendices A.8, A.9, and A.10, the partial derivatives of the i -th swivel angle and cable length with respect to the *EE* pose are expressed as:

$$\frac{\partial \sigma_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{D}(\boldsymbol{\epsilon}), \quad \boldsymbol{\xi}_{\sigma_i} = \frac{1}{\rho_{u_i}} \begin{bmatrix} -\mathbf{w}_i \\ -(\boldsymbol{\rho}_i + \mathbf{d}_i) \times \mathbf{w}_i \end{bmatrix} \quad (2.50)$$

$$\frac{\partial l_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{l_i}^T \mathbf{D}(\boldsymbol{\epsilon}), \quad \boldsymbol{\xi}_{l_i} = \begin{bmatrix} -\mathbf{t}_i \\ -(\mathbf{d}_i + r_i(\mathbf{u}_i + \mathbf{n}_i)) \times \mathbf{t}_i \end{bmatrix} \quad (2.51)$$

In Eq. (2.50), the term $(\boldsymbol{\rho}_i + \mathbf{d}_i)$ represents the relative vector from the platform origin P to the fixed anchor point A_i . Similarly, in Eq. (2.51), the term $(\mathbf{d}_i + r_i(\mathbf{u}_i + \mathbf{n}_i))$ is the relative vector from P to the pulley exit point B_i . The negative signs in both geometric Jacobians reflect the fact that, in this architecture, the pulleys move with the platform toward fixed anchor points.

By stacking these equations for all n cables, we obtain the analytical Jacobians for the deployable architecture:

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\zeta}} = \mathbf{J}_\sigma(\boldsymbol{\zeta}) = \begin{bmatrix} \boldsymbol{\xi}_{\sigma_1}^T \\ \boldsymbol{\xi}_{\sigma_2}^T \\ \vdots \\ \boldsymbol{\xi}_{\sigma_n}^T \end{bmatrix} \mathbf{D}(\boldsymbol{\epsilon}), \quad \frac{\partial \mathbf{l}}{\partial \boldsymbol{\zeta}} = \mathbf{J}_l(\boldsymbol{\zeta}) = \begin{bmatrix} \boldsymbol{\xi}_{l_1}^T \\ \boldsymbol{\xi}_{l_2}^T \\ \vdots \\ \boldsymbol{\xi}_{l_n}^T \end{bmatrix} \mathbf{D}(\boldsymbol{\epsilon}), \quad (2.52)$$

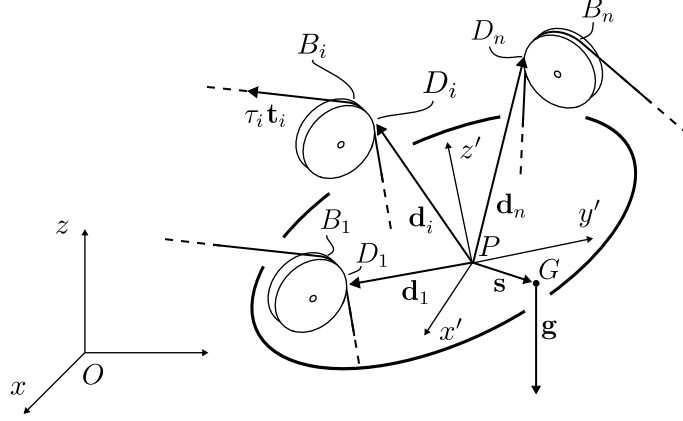
These matrices provide the mapping $\dot{\boldsymbol{\sigma}} = \mathbf{J}_\sigma \dot{\boldsymbol{\zeta}}$ and $\dot{\mathbf{l}} = \mathbf{J}_l \dot{\boldsymbol{\zeta}}$, which are fundamental for the state estimation and self-calibration algorithms discussed in Chapter 4.

2.3.2 Second-order kinematics for deployable *CDPRs*

Similar to the classical case, the second-order kinematics for deployable *CDPRs* is required for robust workspace analysis. The derivation follows the same logic as Section 2.1.2, utilizing the derivatives of the analytical Jacobians with respect to the pose $\boldsymbol{\zeta}$.

The derivatives of the transformation matrix $\mathbf{D}(\boldsymbol{\epsilon})$ are identical to those provided in Eqs. (2.25)–(2.26), as the orientation parametrization remains unchanged. To determine the Hessian of the i -th cable length $\partial^2 l_i / \partial \boldsymbol{\zeta}^2 \in \mathbb{R}^{6 \times 6}$, we differentiate Eq. (2.51) with respect to the pose:

$$\frac{\partial^2 l_i}{\partial \boldsymbol{\zeta}^2} = \frac{\partial \boldsymbol{\xi}_{l_i}^T}{\partial \boldsymbol{\zeta}} \mathbf{D}(\boldsymbol{\epsilon}) + \sum_{j=1}^6 \xi_{l_i,j} \frac{\partial \mathbf{D}_{(j,:)} }{\partial \boldsymbol{\zeta}}, \quad (2.53)$$


 Figure 2.6: *EE* Free-Body Diagram for deployable *CDPRs*

where $\xi_{l_i,j}$ denotes the j -th component of ξ_{l_i} . The derivation $\partial \xi_{l_i}^T / \partial \zeta$ is detailed in A.11 and the term can be written as

$$\frac{\partial \xi_{l_i}^T}{\partial \zeta} = \mathbf{D}^T \bar{\xi}_{l_i}, \quad \bar{\xi}_{l_i} = \xi_{\psi_i} \xi_{n_i}^T + \sin \psi_i \xi_{\sigma_i} \xi_{w_i}^T + (r_i + r_i \cos \psi_i) \xi_{\sigma_i} \xi_{t_i}^T + \mathbf{L}_i^T. \quad (2.54)$$

where

$$\xi_{n_i} = \begin{bmatrix} -\mathbf{n}_i \\ \mathbf{q}_i \times \mathbf{n}_i \end{bmatrix}, \quad \xi_{w_i} = \begin{bmatrix} -\mathbf{w}_i \\ -\mathbf{q}_i \times \mathbf{w}_i \end{bmatrix}, \quad \xi_{t_i} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{t}_i \times \mathbf{w}_i \end{bmatrix}, \quad \mathbf{L}_i = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \tilde{\mathbf{t}}_i \\ \mathbf{0}_{3 \times 3} & -\tilde{\mathbf{t}}_i \tilde{\mathbf{q}}_i + \tilde{\mathbf{q}}_i \tilde{\mathbf{t}}_i \end{bmatrix}, \quad (2.55)$$

2.4 Static model of deployable *CDPRs*

The *CDPR* static model is derived from the mechanical equilibrium of the *EE* (Fig. 2.6). For the purpose of this work, we assume gravity as the only external force acting on the *EE*. Under the assumption that all cables remain taut, the static equilibrium of the *EE* is given by:

$$\Xi^T \boldsymbol{\tau} = \mathbf{f}, \quad \text{with} \quad \mathbf{f} = \begin{bmatrix} \mathbf{g} \\ \mathbf{s} \times \mathbf{g} \end{bmatrix}, \quad (2.56)$$

where $\mathbf{g} = [0, 0, -mg]^T$, m is the platform mass, g is the gravitational acceleration, $\mathbf{s}$ is the position vector from P to the center of mass G , and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the array of cable tensions. Ξ^T is the structure matrix, which relates cable tensions to forces and torques on the *EE*, and is computed as:

$$\Xi^T = [\xi_{l_1}, \dots, \xi_{l_n}]. \quad (2.57)$$

If the *CPDR* is overconstrained ($n > n_d$), Eq. (2.33) admits an infinite number of solutions for $\boldsymbol{\tau}$. Several algorithms have been proposed for solving Eq. (2.33) along with constraints on tension bounds (imposed to avoid cable slackness or breakage) and with optimization of specific performance indexes [64].

2.5 Workspace analysis

The determination of the set of poses that a manipulator can attain, known as its workspace (*WS*), is of primary importance for motion planning [65], geometrical optimiza-

tion [17, 29, 66–68], and the definition of reconfiguration strategies [69–72]. In practical applications, rather than the entire six-dimensional configuration space, specific subsets are typically analyzed [3]. For instance, the *reachable workspace* is the set of positions that the *EE* can reach with at least one orientation [73]. Another relevant subset is the *constant-orientation* (or *translational*) workspace, which comprises all positions reachable by the *EE* for a given, fixed orientation ϵ [43]. The latter is particularly relevant for parallel manipulators, which often perform tasks that require a constant platform attitude.

For *CDPRs*, workspace analysis focuses on identifying poses that satisfy two primary criteria: the absence of mechanical interference (cable-to-cable or cable-to-platform) and wrench feasibility [49, 74–79]. Wrench feasibility is the ability of the robot to balance a prescribed set of external wrenches while maintaining cable tensions within strictly positive and bounded limits. In this work, we focus on static feasibility [49], where the only external wrench considered is the gravitational wrench $\mathbf{f}$ defined in Eqs. (2.33) and (2.56).

The set of statically feasible poses is the Wrench-Feasible Workspace (*WFW*), which can be formally defined as:

$$WFW = \{\zeta \in \mathbb{R}^6 \mid \Xi_l^T(\zeta)\boldsymbol{\tau} = \mathbf{f}(\zeta), \quad \tau_{min} \leq \boldsymbol{\tau} \leq \tau_{max}\}, \quad (2.58)$$

where $\Xi_l^T(\zeta)$ is the transposed geometric Jacobian matrix (often called the structure matrix) and the symbol $\leq$ denotes an element-wise inequality. The positive bounds τ_{min} and τ_{max} are imposed to prevent cable slackness and mechanical failure, respectively. The method for evaluating whether a pose belongs to this set depends fundamentally on the robot's degree of constraint.

2.5.1 Overconstrained CDPRs and the Tension Distribution Problem

In *OCDPRs*, the number of cables exceeds the *EE* degrees of freedom ($n > n_d$), making the inverse static problem underdetermined. The structure matrix Ξ_l^T has a non-trivial null space, which implies that for a given pose, there are infinite tension distributions $\boldsymbol{\tau}$ that satisfy equilibrium. A pose is included in the *WFW* if there exists at least one vector $\boldsymbol{\tau}$ such that:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_p + \Xi_l^{T\perp} \boldsymbol{\lambda}, \quad \tau_{min} \leq \boldsymbol{\tau} \leq \tau_{max}, \quad (2.59)$$

where $\boldsymbol{\tau}_p = \Xi_l^{-T} \mathbf{f}$ is the particular solution of the static equilibrium (e.g., the minimum-norm solution if Ξ_l^{-T} is the pseudo inverse of Ξ_l^T), $\Xi_l^{T\perp}$ is the matrix spanning the null space of Ξ_l^T , and $\boldsymbol{\lambda}$ is a vector of arbitrary coefficients. Determining the feasibility of a pose is therefore equivalent to solving the Tension Distribution (*TD*) problem, i.e., finding a $\boldsymbol{\lambda}$ that shifts the tensions into the feasible range $[\tau_{min}, \tau_{max}]$ while optimizing a specific performance criterion [34–38, 64, 80–86]. In Chapter 3, we investigate how these distributions can be selected to ensure robustness against modeling and measurement errors.

2.5.2 Underactuated CDPRs and the Inverse Geometrico-Static Problem

For *UACDPRs* ($n < n_d$), the inverse static problem is overdetermined. Therefore, generally, there is no cable tension array $\boldsymbol{\tau}$ that satisfies the static equilibrium (Eq. (2.33)) or

(2.56)) for an arbitrary given pose ζ . In this case, the wrench-feasibility problem (2.60) has to be solved considering the kinematic relations (2.13) (or equivalently (2.48)) and imposing only a subset of n coordinates of the EE pose:

$$\begin{cases} -\Xi_l^T \boldsymbol{\tau} + \mathbf{f} = \mathbf{0}_{6 \times 1} \\ \boldsymbol{\rho}_1^T \boldsymbol{\rho}_1 - (l_1 - r_1(\pi - \psi_1))^2 = 0 \\ \vdots \\ \boldsymbol{\rho}_n^T \boldsymbol{\rho}_n - (l_n - r_n(\pi - \psi_n))^2 = 0 \end{cases}. \quad (2.60)$$

The latter is the so-called Inverse Geometrico-Static Problem (*IGSP*), whose solution comprises the cable lengths, cable tensions, and the remaining $(n_d - n)$ EE coordinates. In [49], the authors proposed a modified *IGSP* that always considers the position as the imposed subset of the EE pose, and the orientation is found so that, at the equilibrium, a desired tension distribution is achieved, when possible: Following the approach described in [49], the modified inverse geometrico-static problem consists of determining the cable tensions and the EE orientation for a given EE position within the workspace, under a fixed tension distribution strategy (such as the barycentric method [87]). This leads to solving, at each point, a system of three nonlinear equations in three unknowns (namely the EE orientation):

$$(\Xi_M^T \Xi_F^{-T} - \tilde{\mathbf{s}}') \mathbf{g} + \Xi_M^T \Xi_F^{-T \perp} \boldsymbol{\lambda} = \mathbf{0}_{3 \times 1}, \quad (2.61)$$

where $\Xi_F^T \in \mathbb{R}^{3 \times n}$ and $\Xi_M^T \in \mathbb{R}^{3 \times n}$ are the translational and rotational blocks of the structure matrix Ξ_l^T , $\Xi_F^{-T} \in \mathbb{R}^{n \times 3}$ is any right inverse of Ξ_F^T and $\Xi_F^{T \perp} \in \mathbb{R}^{m \times (n-3)}$ spans the nullspace of Ξ_F^T , $\boldsymbol{\lambda} \in \mathbb{R}^{(n-3) \times 1}$ is the array derived by the chosen tension distribution algorithm. Then, cable tensions are obtained as:

$$\boldsymbol{\tau} = \Xi_F^{-T} \mathbf{g} + \Xi_F^{T \perp} \boldsymbol{\lambda}. \quad (2.62)$$

If the tensions satisfy the inequality constraints

$$\tau_{min} \leq \boldsymbol{\tau} \leq \tau_{max} \quad (2.63)$$

and the pose is statically stable, then the pose is included in the static workspace. Static stability is assessed by verifying the positive definiteness of the free-motion stiffness matrix $\mathbf{K}_f^\perp$ [88]:

$$\mathbf{K}_f^\perp \triangleq \Xi^{\perp T} (\mathbf{K} + \mathbf{Q}) \Xi^\perp \quad (2.64)$$

where $\Xi^{\perp T} \in \mathbb{R}^{(6-n) \times 6}$ is any left nullspace basis of Ξ^T ($\Xi^{\perp T} \Xi^T = \mathbf{0}_{(6-n) \times n}$) and the matrices $\mathbf{K}$ and $\mathbf{Q}$ are defined for classic *CDPRs* as:

$$\begin{aligned} \mathbf{K} &\triangleq \sum_{i=1}^n \tau_i \begin{bmatrix} \mathbf{T}_i & -\mathbf{T}_i \tilde{\mathbf{a}}_i \\ \tilde{\mathbf{a}}_i \mathbf{T}_i & -\tilde{\mathbf{a}}_i \mathbf{T}_i \tilde{\mathbf{a}}_i \end{bmatrix} + \sum_{i=1}^n \tau_i \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{t}}_i \tilde{\mathbf{a}}_i \end{bmatrix}, \\ \mathbf{T}_i &\triangleq \frac{\|\mathbf{u}_i \times \mathbf{n}_i\| \mathbf{w}_i \mathbf{w}_i^T}{\mathbf{u}_i \cdot (\mathbf{a}_i - \mathbf{d}_i)} + \frac{\mathbf{n}_i \mathbf{n}_i^T}{\|\boldsymbol{\rho}_i\|}, \quad \mathbf{Q} \triangleq \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{s}} \tilde{\mathbf{g}} \end{bmatrix}. \end{aligned} \quad (2.65)$$

2.5.3 Computational Implementation

When analyzing the constant orientation static *WS* for *OCDPRs* or the static *WS* for *UACDPRs* using the modified *IGSP*, the installation volume is typically discretized into

a 3D grid of points (voxels), equidistant along the principal axes. Each voxel center is evaluated for wrench feasibility, creating a point-cloud representation of the workspace [43,49]. While boundary exploration algorithms can reduce computational time by focusing on transitions between feasible and infeasible regions [89–91], voxel-based discretization remains a robust method for evaluating global performance indices. In this work, a trade-off between mesh resolution and computational demand is selected to ensure accurate boundary determination for both *OCDPR* and *UACDPR* architectures.

Chapter 3

Control-based workspace analysis of *OCDPRs*

Motion control of *OCDPRs* can be obtained through several techniques. Feedback Linearization [39, 92], Sliding Mode Control [37, 40], Model Inversion with force correction [93], Model Predictive Control [41, 42, 94], Hybrid Position/Force Control in cartesian space [83, 95]. The control input of the aforementioned methods, except [93], is often the motor torque or the cable tension. These quantities are not always available as direct low-level inputs, but they require a conversion to motor speed or motor position through an additional regulator, which needs additional tuning. Moreover, motor torque or cable tension are indirect measures that require a model characterization and calibration, which, in turn, may introduce a non-negligible error.

One effective alternative to limit the error associated with the control input is the Hybrid Position/Force Control in joint-space (*HC*) [43, 44]. In this method, a number of cables equal to the *EE* DoFs are length-controlled, while the remaining cables, corresponding to the degree of redundancy, are force-controlled. The length-controlled cables are responsible for enforcing the *EE* pose, while the force-controlled cables manage the tension distribution, acting as servo-controlled springs. Indeed, a peculiarity of this regulator is that the problems of motion control and tension distribution are decoupled, making its design significantly easier than the alternatives. When *HC* is employed, the choice of length- and force-controlled cables is of primary importance because it influences the motion and *TD* performances of the robot. Moreover, if the control strategy, i.e., the combination of length- and force-controlled cables, doesn't change during task execution, the robot can be designed with only those sensors that are necessary for the regulator, thus reducing the hardware costs.

Within the *WFW*, performance indexes can be evaluated to select the optimal control strategy that ensures the best results in terms of performance. For example, Zake et al. in [46] define a Control Stability Workspace to assess the visual servoing controller stability in the presence of perturbations. Regarding *HC*, Mattioni et al. [43] used a performance index called force-distribution sensitivity to cable-tension errors to evaluate the maximum tension error caused on length-controlled cables while force-controlling the others, and proposed a heuristic method for selecting the optimal combination of length- and force-controlled cables [45]. However, the force-distribution sensitivity doesn't take into account any influence of the cable-length errors on the *TD* or any accuracy performance.

In this chapter¹ the *HC* formulation is presented in Sec. 3.1. Then, the *Wrench-Feasible Error-Insensitive Workspace* (*WFEIW*), a theoretical tool to evaluate accuracy performance, and control-error sensitivity in hybrid-controlled *OCDPRs*, is introduced (Sec. 3.2). In the *WFEIW*, user-defined accuracy requirements and cable tensions within predetermined bounds are guaranteed, accounting for sensor-error expectations. The *WFEIW* may be significantly smaller compared to the ideal *WFW*, thus limiting the actual *OCDPR* motion capabilities.

The *WFEIW* is analyzed for several *OCDPR* architectures to illustrate how design choices impact the shape and dimensions of the usable workspace. In section 3.3, an algorithm for optimal control-strategy selection and design optimization based on the *WFEIW* analysis is presented. The objective is thus to provide a theoretical tool for *OCDPR* design and control optimization based on *WFEIW* analysis, to potentially extend the actual range of motion of the *EE*. In particular, the two objectives of the optimization algorithm in Sec. 3.3 are: (i) the determination of the most suitable combination of length- and force-controlled cables and (ii) the determination of the platform's geometry (namely the relative position of the cable attachment points on the *EE*) that maximizes the *WFEIW* volume. The former problem is a discrete one, as its unknowns are discrete sets, while the latter is continuous. Since the results of the two optimizations are interdependent, a nested optimization problem is implemented where exhaustive search and gradient-based optimization algorithms are used independently. Several simulations are conducted on well-known *OCDPR* architectures as application examples of the control-strategy and design optimization algorithm.

3.1 Hybrid Control

In hybrid position-force control in the joint space, n_d cables are length-controlled, aiming at regulating the *EE* pose, and $\mu = n - n_d$ cables are force-controlled, aiming at managing the *TD*. Without loss of generality, we assume that the first n_d cables are length-controlled and the last μ cables are force-controlled. Accordingly, the tension array and the structure matrix can be partitioned as:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_d \\ \boldsymbol{\tau}_c \end{bmatrix}, \quad \boldsymbol{\Xi}_l^T = [\boldsymbol{\Xi}_d \quad \boldsymbol{\Xi}_c] \quad (3.1)$$

where $\boldsymbol{\tau}_c \in \mathbb{R}^\mu$ is the tension array of force-controlled cables, $\boldsymbol{\tau}_d \in \mathbb{R}^{n_d}$ is the tension array of length-controlled cables, and $\boldsymbol{\Xi}_d \in \mathbb{R}^{n_d \times n_d}$ and $\boldsymbol{\Xi}_c \in \mathbb{R}^{n_d \times \mu}$ are the corresponding blocks of the structure matrix.

Recalling Eq. (2.59), for *HC*, to decouple the contribution of length- and force-controlled cables, a specific choice of the inverse and nullspace is adopted:

$$\boldsymbol{\tau} = \boldsymbol{\Xi}_{hc}^{-T} \mathbf{f} + \boldsymbol{\Xi}_{hc}^{T\perp} \boldsymbol{\tau}_c \quad (3.2)$$

where:

$$\boldsymbol{\Xi}_{hc}^{-T} = \begin{bmatrix} \boldsymbol{\Xi}_d^{-1} \\ \mathbf{0}_{\mu \times n_d} \end{bmatrix}, \quad \boldsymbol{\Xi}_{hc}^{T\perp} = \begin{bmatrix} -\boldsymbol{\Xi}_d^{-1} \boldsymbol{\Xi}_c \\ \mathbf{I}_{\mu \times \mu} \end{bmatrix} \quad (3.3)$$

with $\mathbf{0}_{\mu \times n_d}$ and $\mathbf{I}_{\mu \times \mu}$ being $\mu \times n_d$ and $\mu \times \mu$ null and identity matrices. This formulation ensures that regardless of the adopted *TD* algorithm, $\boldsymbol{\tau}$ can be regulated by only controlling the tension of μ cables, $\boldsymbol{\tau}_c$ while using n_d cables to regulate the *EE* pose [43].

¹The content of this chapter is partially based on [47, 48] submitted to *Springer Proceedings in Advanced Robotics and Mechanism and Machine Theory*.

Moreover, expressing the nullspace basis $\Xi_{hc}^{T\perp}$ as in Eq. (3.3) allows to obtain a linear dependency of the overall tension distribution $\boldsymbol{\tau}$ on the tension of force-controlled cables $\boldsymbol{\tau}_c$. Thanks to Eqs. (3.1), (3.2) and (3.3), the relation between length- and force-controlled cables is:

$$\boldsymbol{\tau}_d = \Xi_d^{-1} (\mathbf{f} - \Xi_c \boldsymbol{\tau}_c) \quad (3.4)$$

3.2 Wrench-Feasible Error-Insensitive Workspace

One of the primary goals in robot motion control is achieving accurate position and orientation of the *EE*. Accuracy can be evaluated through kinematic sensitivities, by analyzing how the *EE*-pose errors are affected by actuator displacement errors. In [96], orientational and positional sensitivities to errors in joint variables are best evaluated through a mixed-norm approach (infinity and 2-norm). However, in this work, the classical kinematic indices proposed in [97] are adopted, because they retain physical meaning and require less computational effort. Indeed, the infinity norm is sufficient to determine the upper bounds of the kinematic indices, since it represents the worst-case sensitivity.

From Sec. 2.1.1, the kinematic relation between the infinitesimal variation of cable lengths and the infinitesimal variation of the *EE* pose can be expressed as $d\mathbf{l} = \mathbf{J}d\boldsymbol{\zeta}$. Using the same partition of the tension array shown in Eq. (3.1), the cable length array $\mathbf{l}$ and the analytical jacobian $\mathbf{J}$ can be expressed as:

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_d \\ \mathbf{l}_c \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_d^T \\ \mathbf{J}_c^T \end{bmatrix} \quad (3.5)$$

where $\mathbf{l}_d \in \mathbb{R}^{n_d}$ and $\mathbf{l}_c \in \mathbb{R}^\mu$ denote the length-controlled and force-controlled cable lengths, respectively, with $\mathbf{J}_d \in \mathbb{R}^{n_d \times n_d}$ and $\mathbf{J}_c \in \mathbb{R}^{n_d \times \mu}$. As long as cables are inextensible, force-controlled cables are not used for *EE* pose control, and thus their lengths $\mathbf{l}_c$ do not influence the pose variation. Therefore, the *EE*-pose only depends on the lengths $\mathbf{l}_d$ of length-controlled cables, allowing the infinity-norm displacement sensitivity σ_∞^{p,l_d} and rotational sensitivity $\sigma_\infty^{\epsilon,l_d}$ to cable-length unit errors to be defined as:

$$\sigma_\infty^{p,l_d} \triangleq \max_{\|\mathbf{d}\mathbf{l}_d\|_\infty=1} \|\mathbf{d}\mathbf{p}\|_\infty \quad \sigma_\infty^{\epsilon,l_d} \triangleq \max_{\|\mathbf{d}\mathbf{l}_d\|_\infty=1} \|\mathbf{d}\boldsymbol{\epsilon}\|_\infty \quad (3.6)$$

The use of the infinity norm to limit the actuator displacements is appropriate, since the elements of the array $\|\mathbf{d}\mathbf{l}_d\|$ are independent from each other and the infinity norm allows each of them to be comprised within ± 1 [98]. The use of the infinity norm to evaluate the magnitude of the positional and orientational errors $\mathbf{d}\mathbf{p}$ and $\mathbf{d}\boldsymbol{\epsilon}$ is, instead, questionable, since the components of these vectors are not independent, and the use of the 2-norm would be better motivated, as shown in [96]. However, in this work, the classical kinematic indices proposed in [97], making use of infinity norms only, are adopted, because they retain physical meaning and require less computational effort. Indeed, the infinity norm is sufficient to determine the upper bounds of the kinematic indices, since it represents the worst-case sensitivity.

The sensitivities in Eq. (3.6) can be computed by solving the following linear programming problems [97]:

$$\sigma_\infty^{p,l_d} = \max_{i=1,2,3} \left(\max_{d\boldsymbol{\zeta}} \mathbf{e}_i^T d\boldsymbol{\zeta} \quad \text{s.t.} \quad \mathbf{L}_d d\boldsymbol{\zeta} - \mathbf{1}_{2n_d \times 1} \leq \mathbf{0}_{2n_d \times 1} \right) \quad (3.7)$$

$$\sigma_{\infty}^{r,l_d} = \max_{i=4,5,6} \left(\max_{d\zeta} \mathbf{e}_i^T d\zeta \quad \text{s.t.} \quad \mathbf{L}_d d\zeta - \mathbf{1}_{2n_d \times 1} \leq \mathbf{0}_{2n_d \times 1} \right) \quad (3.8)$$

where $\mathbf{e}_i \in \mathbb{R}^{n_d}$ is an array composed of zeros except for the i -th element that is equal to 1, $\mathbf{1}_{2n_d \times 1}$ is an array of size $2n_d$ in which all elements are equal to 1, $\mathbf{0}_{2n_d \times 1}$ is an array of size $2n_d$ in which all elements are equal to 0, and $\mathbf{L}_d = [\mathbf{J}_d - \mathbf{J}_d]^T$.

After pose accuracy, cable-tension management is the second relevant problem to be addressed in *OCDPRs*. Evaluating the *TD* sensitivity to actuator errors when *HC* is employed is an effective analysis and synthesis tool for control optimization. The *TD* sensitivity to tension unit errors in force-controlled cables, $\sigma_{\infty}^{\tau, \tau_c}$, was introduced in [43], while the *TD* sensitivity to cable-length unit errors $\sigma_{\infty}^{\tau, l_d}$, was introduced in [49]:

$$\sigma_{\infty}^{\tau, \tau_c} \triangleq \max_{\|d\tau_c\|_{\infty}=1} \|d\tau\|_{\infty} \quad \sigma_{\infty}^{\tau, l_d} \triangleq \max_{\|dl_d\|_{\infty}=1} \|d\tau\|_{\infty} \quad (3.9)$$

From Eq.(3.2), the *TD* sensitivity to tension errors can be computed as:

$$\sigma_{\infty}^{\tau, \tau_c} = \left\| \frac{\partial \tau}{\partial \tau_c} \right\|_{\infty} = \|\Xi_{hc}^{T\perp}\|_{\infty} \quad (3.10)$$

Since τ_c does not depend on $\mathbf{l}_d$ the *TD* sensitivity to cable-length errors can be computed as:

$$\sigma_{\infty}^{\tau, l_d} = \left\| \frac{\partial \tau}{\partial \mathbf{l}_d} \right\|_{\infty} = \left\| \frac{\partial \tau_d}{\partial \mathbf{l}_d} \right\|_{\infty} \quad (3.11)$$

Using the chain rule to explicit the pose dependency, the derivative of Eq. (3.11) becomes:

$$\frac{\partial \tau_d}{\partial \mathbf{l}_d} = \frac{\partial \tau_d}{\partial \zeta} \frac{\partial \zeta}{\partial \mathbf{l}_d} \quad (3.12)$$

Considering the partitions of Eq. (3.5), the first order kinematics $\dot{\mathbf{l}} = \mathbf{J}_l \dot{\zeta}$ can be expressed as:

$$\begin{bmatrix} \frac{\partial \mathbf{l}_d}{\partial t} \\ \frac{\partial \mathbf{l}_c}{\partial t} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_d^T \\ \mathbf{J}_c^T \end{bmatrix} \frac{\partial \zeta}{\partial t} \quad (3.13)$$

The upper block of Eq. (3.13), namely

$$\frac{\partial \mathbf{l}_d}{\partial t} = \mathbf{J}_d^T \frac{\partial \zeta}{\partial t} \quad (3.14)$$

allows the second derivative in the right-hand side of Eq. (3.12) to be expressed as:

$$\frac{\partial \zeta}{\partial \mathbf{l}_d} = \mathbf{J}_d^{-T} \quad (3.15)$$

where the square matrix $\mathbf{J}_d \in \mathbb{R}^{n_d \times n_d}$ is assumed to be non-singular. The first derivative in the right-hand side of Eq. (3.12) can be expanded from Eq. (3.4) as:

$$\frac{\partial \tau_d}{\partial \zeta} = \frac{\partial \Xi_d^{-1}}{\partial \zeta} \mathbf{f} + \Xi_d^{-1} \frac{\partial \mathbf{f}}{\partial \zeta} - \frac{\partial \Xi_d^{-1}}{\partial \zeta} \Xi_c \tau_c - \Xi_d^{-1} \frac{\partial \Xi_c}{\partial \zeta} \tau_c \quad (3.16)$$

in which the term $\frac{\partial \tau_c}{\partial \zeta}$ was omitted because the tension in force-controlled cables is an independent variable. Exploiting the relation $\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{q}} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \mathbf{q}} \mathbf{A}^{-1}$ in which $\mathbf{A}$ and $\mathbf{q}$ are a generic invertible matrix and a generic vector, (3.16) can be written as:

$$\frac{\partial \tau_d}{\partial \zeta} = -\Xi_d^{-1} \frac{\partial \Xi_d}{\partial \zeta} \Xi_d^{-1} \mathbf{f} + \Xi_d^{-1} \frac{\partial \mathbf{f}}{\partial \zeta} + \Xi_d^{-1} \frac{\partial \Xi_d}{\partial \zeta} \Xi_d^{-1} \Xi_c \tau_c - \Xi_d^{-1} \frac{\partial \Xi_c}{\partial \zeta} \tau_c \quad (3.17)$$

and thus, by collecting $-\Xi_d^{-1}$:

$$\frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\zeta}} = -\Xi_d^{-1} \left(\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \mathbf{f} - \frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \Xi_c \boldsymbol{\tau}_c + \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_c - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \quad (3.18)$$

The first two terms in the parenthesis of Eq. (3.18) can be collected through Eq. (3.4) in a single term, obtaining:

$$\frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\zeta}} = -\Xi_d^{-1} \left(\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_d + \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_c - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \quad (3.19)$$

Substituting Eqs. (3.19) and (3.15) in (3.12) yields:

$$\frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} = -\Xi_d^{-1} \left(\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_d + \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_c - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \quad (3.20)$$

and since $\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} = \begin{bmatrix} \frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} & \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \end{bmatrix}$ and $\boldsymbol{\tau} = [\boldsymbol{\tau}_d^T \quad \boldsymbol{\tau}_c^T]^T$, Eq. (3.20) can be finally expressed as:

$$\frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} = -\Xi_d^{-1} \left(\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau} - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \quad (3.21)$$

where $\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}$ and $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}}$ can be computed as follows²:

$$\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau} = \sum_{i=1}^n \tau_i \frac{\partial \xi_{l_i}}{\partial \boldsymbol{\zeta}}, \quad \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & m \tilde{\mathbf{g}} \tilde{\mathbf{s}} \end{bmatrix} \mathbf{D} \quad (3.22)$$

which yields

$$\sigma_{\infty}^{\tau, l_d} = \left\| -\Xi_d^{-1} \left(\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau} - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \right\|_{\infty} \quad (3.23)$$

Equation (3.23) shows that the *TD* sensitivity to cable-length errors depends on $\boldsymbol{\tau}$, and thus on $\boldsymbol{\tau}_c$. This implies that $\sigma_{\infty}^{\tau, l_d}$ depends not only on the *EE* pose, like common accuracy performance indices, but also on the specific choice of the *TD* algorithm used to compute $\boldsymbol{\tau}_c$. This result poses a problem, since workspace analysis should ideally be independent of any specific *TD* algorithm. This issue is addressed in the following Section 3.2.1.

3.2.1 WFEIW Analysis

The following workspace analysis is performed on the translational workspace to allow a better visualization and interpretation of the obtained results. However, the analysis method and the workspace optimization introduced in Sec. 3.3, can be generally applied to any set of configurations reachable by the *EE*. The robot installation volume is discretized in a grid of points, assuming a constant *EE* orientation, and checking every point for inclusion in the workspace under consideration. The wrench feasibility condition can be verified by checking the existence of a convex polygon bounded by the inequalities (2.59) in the controlled-tension space, as demonstrated in [43] by using the algorithm of Guttefarde et al. [36]. Additionally, cable-collision algorithms [78] can be used to prune geometrically unfeasible configurations.

²The full demonstration of these expressions can be found in Eqs. (33) and (39) of [88].

Then, poses within the *WFW* that are sensitive to tension variations due to control or sensor errors can be identified and called Tension Sensitive (*TS*) poses. When *HC* is used, tension errors result from both cable-length errors $\Delta \mathbf{l}_d$ and tension errors $\Delta \boldsymbol{\tau}_c$. These errors can be incorporated into the constraints of Equation (2.59), by performing its Taylor expansion:

$$\tau_{min} \leq \boldsymbol{\tau}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} \Delta \mathbf{l}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} \Delta \boldsymbol{\tau}_c \leq \tau_{max} \quad (3.24)$$

$$\tau_{min} \leq \boldsymbol{\tau}_c + \Delta \boldsymbol{\tau}_c \leq \tau_{max} \quad (3.25)$$

The term $\frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c}$ can be computed from Eq. (3.4) as:

$$\frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} = -\Xi_d^{-1} \Xi_c \quad (3.26)$$

Substituting Eqs. (3.20) and (3.26), the Taylor expansion of $\boldsymbol{\tau}_d$ becomes:

$$\begin{aligned} \boldsymbol{\tau}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} \Delta \mathbf{l}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} \Delta \boldsymbol{\tau}_c = & \boldsymbol{\tau}_d - \Xi_d^{-1} \left(\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \mathbf{f} - \right. \\ & \left. - \frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \Xi_c \boldsymbol{\tau}_c + \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_c - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \Delta \mathbf{l}_d - \Xi_d^{-1} \Xi_c \Delta \boldsymbol{\tau}_c \end{aligned} \quad (3.27)$$

The terms in the right-hand side of Eq. (3.27) that do not depend on $\boldsymbol{\tau}_c$ can be collected into the array $\Delta \boldsymbol{\tau}_e$:

$$\Delta \boldsymbol{\tau}_e(\boldsymbol{\zeta}, \Delta \mathbf{l}_d, \Delta \boldsymbol{\tau}_c) = -\Xi_d^{-1} \left(\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \mathbf{f} - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \Delta \mathbf{l}_d - \Xi_d^{-1} \Xi_c \Delta \boldsymbol{\tau}_c \quad (3.28)$$

so that Eq. (3.27) can be expressed as:

$$\boldsymbol{\tau}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} \Delta \mathbf{l}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} \Delta \boldsymbol{\tau}_c = \boldsymbol{\tau}_d + \Delta \boldsymbol{\tau}_e - \Xi_d^{-1} \left(-\frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} \Xi_d^{-1} \Xi_c \boldsymbol{\tau}_c + \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \boldsymbol{\tau}_c \right) \mathbf{J}_d^{-T} \Delta \mathbf{l}_d \quad (3.29)$$

Considering that $\frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} = \begin{bmatrix} \frac{\partial \Xi_d}{\partial \boldsymbol{\zeta}} & \frac{\partial \Xi_c}{\partial \boldsymbol{\zeta}} \end{bmatrix}$ and using the definition of matrix $\Xi_{hc}^{T\perp}$ in Eq. (3.3), Eq. (3.29) becomes:

$$\boldsymbol{\tau}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} \Delta \mathbf{l}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} \Delta \boldsymbol{\tau}_c = \boldsymbol{\tau}_d + \Delta \boldsymbol{\tau}_e - \Xi_d^{-1} \frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \Xi_{hc}^{T\perp} \boldsymbol{\tau}_c \mathbf{J}_d^{-T} \Delta \mathbf{l}_d \quad (3.30)$$

Taking the last term of Eq. (3.30) and using the properties of the tensor product, we can define a matrix $\mathbf{B}$ that depends only on the *EE* pose, obtaining a linear relation on $\boldsymbol{\tau}_c$, through the following passages:

$$\begin{aligned} -\Xi_d^{-1} \frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \Xi_{hc}^{T\perp} \boldsymbol{\tau}_c \mathbf{J}_d^{-T} \Delta \mathbf{l}_d &= -\Xi_d^{-1} \left(\sum_{i=1}^n \frac{\partial \xi_{l_i}^T}{\partial \boldsymbol{\zeta}} (\Xi_{hc}^{T\perp} \boldsymbol{\tau}_c)_i \right) \mathbf{J}_d^{-T} \Delta \mathbf{l}_d = \\ &= -\Xi_d^{-1} \sum_{j=1}^m \sum_{i=1}^n \frac{\partial \xi_{l_i}^T}{\partial \zeta_j} (\Xi_{hc}^{T\perp} \boldsymbol{\tau}_c)_i (\mathbf{J}_d^{-T} \Delta \mathbf{l}_d)_j = \\ &= -\Xi_d^{-1} \sum_{i=1}^n \sum_{j=1}^m \frac{\partial \xi_{l_i}}{\partial \zeta_j} (\mathbf{J}_d^{-T} \Delta \mathbf{l}_d)_j (\Xi_{hc}^{T\perp} \boldsymbol{\tau}_c)_i = \\ &= -\Xi_d^{-1} \frac{\partial \Xi_l^T}{\partial \boldsymbol{\zeta}} \mathbf{J}_d^{-T} \Delta \mathbf{l}_d \Xi_{hc}^{T\perp} \boldsymbol{\tau}_c = \\ &= \mathbf{B}(\boldsymbol{\zeta}, \Delta \mathbf{l}_d) \boldsymbol{\tau}_c \end{aligned} \quad (3.31)$$

where $(\cdot)_i$ and $(\cdot)_j$ denote, respectively, the i -th and the j -th element of the vector in parentheses.

Substituting Eq. (3.31) in Eq. (3.30), gives:

$$\boldsymbol{\tau}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{l}_d} \Delta \mathbf{l}_d + \frac{\partial \boldsymbol{\tau}_d}{\partial \boldsymbol{\tau}_c} \Delta \boldsymbol{\tau}_c = \boldsymbol{\tau}_d + \Delta \boldsymbol{\tau}_e(\boldsymbol{\zeta}, \Delta \mathbf{l}_d, \Delta \boldsymbol{\tau}_d) + \mathbf{B}(\boldsymbol{\zeta}, \Delta \mathbf{l}_d) \boldsymbol{\tau}_c \quad (3.32)$$

By taking advantage of the previous manipulations, the disequalities (3.24) take the form:

$$\tau_{min} \leq \boldsymbol{\tau}_d + \Delta \boldsymbol{\tau}_e + \mathbf{B} \boldsymbol{\tau}_c \leq \tau_{max} \quad (3.33)$$

where:

$$\Delta \boldsymbol{\tau}_e(\boldsymbol{\zeta}, \Delta \mathbf{l}_d, \Delta \boldsymbol{\tau}_c) = -\boldsymbol{\Xi}_d^{-1} \left(\frac{\partial \boldsymbol{\Xi}_d}{\partial \boldsymbol{\zeta}} \boldsymbol{\Xi}_d^{-1} \mathbf{f} - \frac{\partial \mathbf{f}}{\partial \boldsymbol{\zeta}} \right) \mathbf{J}_d^{-T} \Delta \mathbf{l}_d - \boldsymbol{\Xi}_d^{-1} \boldsymbol{\Xi}_c \Delta \boldsymbol{\tau}_c \quad (3.34)$$

$$\mathbf{B}(\boldsymbol{\zeta}, \Delta \mathbf{l}_d) = -\boldsymbol{\Xi}_d^{-1} \frac{\partial \boldsymbol{\Xi}_d^T}{\partial \boldsymbol{\zeta}} \mathbf{J}_d^{-T} \Delta \mathbf{l}_d \boldsymbol{\Xi}_{hc}^{T\perp} \quad (3.35)$$

Here, the error vectors $\Delta \mathbf{l}_d$ and $\Delta \boldsymbol{\tau}_c$ represent the sensor uncertainty or the control errors with their largest values being $\Delta \tau_{c,i} = \pm \delta \tau$ for $i = 1, \dots, \mu$ where $\Delta l_{d,j} = \pm \delta l$ for $j = 1, \dots, n_d$, and δl and $\delta \tau$ are the sensor or control errors on cable lengths or tensions respectively. Equations (3.25) and (3.33) are linear with respect to $\boldsymbol{\tau}_c$, thus traditional algorithms for wrench feasibility can be used to verify the feasibility of a generic pose $\boldsymbol{\zeta}$, and compute a feasible TD [33]. In fact, substituting Eq. (3.2) in Eqs. (3.25) and (3.33) results in:

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\Xi}_{hc}^{-T} \mathbf{f} + \begin{bmatrix} \Delta \boldsymbol{\tau}_e \\ \Delta \boldsymbol{\tau}_c \end{bmatrix} + \left(\boldsymbol{\Xi}_{hc}^{T\perp} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{\mu \times \mu} \end{bmatrix} \right) \boldsymbol{\tau}_c \leq \boldsymbol{\tau}_{max} \quad (3.36)$$

where $\boldsymbol{\tau}_{min} \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{max} \in \mathbb{R}^n$ are vectors in which all elements are equal to the minimum and the maximum tension limit, respectively. Isolating the term dependent on $\boldsymbol{\tau}_c$ in Eq. (3.36), and explicitly highlighting the dependency from $\Delta \mathbf{l}_d$ and $\Delta \boldsymbol{\tau}_c$ we have:

$$\boldsymbol{\tau}_{min} - \boldsymbol{\Xi}_{hc}^{-T} \mathbf{f} - \begin{bmatrix} \Delta \boldsymbol{\tau}_e(\Delta \mathbf{l}_d, \Delta \boldsymbol{\tau}_c) \\ \Delta \boldsymbol{\tau}_c \end{bmatrix} \leq \left(\boldsymbol{\Xi}_{hc}^{T\perp} + \begin{bmatrix} \mathbf{B}(\Delta \mathbf{l}_d) \\ \mathbf{0}_{\mu \times \mu} \end{bmatrix} \right) \boldsymbol{\tau}_c \leq \boldsymbol{\tau}_{max} - \boldsymbol{\Xi}_{hc}^{-T} \mathbf{f} - \begin{bmatrix} \Delta \boldsymbol{\tau}_e(\Delta \mathbf{l}_d, \Delta \boldsymbol{\tau}_c) \\ \Delta \boldsymbol{\tau}_c \end{bmatrix} \quad (3.37)$$

There are 2^n possible combinations of elements in vectors $\Delta \mathbf{l}_d$ and $\Delta \boldsymbol{\tau}_c$, emerging from the choice of the + or – sign in each element. Every combination must be considered, and if any one of them results in an infeasible solution, the pose is considered *TS*. The *WFW* without the *TS* region is called the Tension Error-Insensitive (*TEI*) Workspace.

We can also consider the effect of control or sensor errors on the positioning and orientation accuracy. The kinematic sensitivities introduced in Sec. (3.7), and (3.8), quantify the impact of control input or joint-variable measurement errors on the *EE* pose. When sensitivities are large, even slight variations in control inputs can result in significant deviations from the desired *EE* pose. Thus, to ensure precision within the *WFW*, upper limits on kinematic sensitivities can be imposed in the form:

$$\sigma_{\infty}^{Pl_d} \leq \sigma_{\infty, max}^{Pl_d}, \quad \sigma_{\infty}^{\epsilon, l_d} \leq \sigma_{\infty, max}^{\epsilon, l_d} \quad (3.38)$$

where $\sigma_{\infty, max}^{Pl_d}$ and $\sigma_{\infty, max}^{\epsilon, l_d}$ denote the maximum allowable sensitivities. Their value can be decided according to task requirement, robot hardware, control responsiveness, and desired precision. If a pose within the *TEI* workspace does not satisfy the constraints set in Eq. (3.38), the pose is considered Kinematic Sensitive (*KS*), and the desired accuracy is not ensured. The *TEI* workspace without the *KS* region is called

Wrench-Feasible Error-Insensitive (*WFEI*) workspace (or *WFEIW*). When a hybrid controller is employed for motion regulation within the *WFEI* workspace, the wrench feasibility is expected to remain unaffected by cable-tension and cable-length errors. Furthermore, user-defined kinematic accuracy requirements are likely to be met throughout this region.

3.2.2 Application examples

As application examples of the workspace analysis tool developed in Section 3.2.1, three 8-cable *OCDPR* architectures are considered: the *IPanema 3* (developed by the Fraunhofer Institute for Manufacturing and Automation) represented in Fig. (3.1), the *HRP - Cable Robot* (a prototype at LIRMM, CNRS - University of Montpellier) represented in Fig (3.2) and the *IRMA-8* deployable prototype (IRMAL@B - University of Bologna) shown in Fig. (3.3). The geometric parameters for the *IPanema 3*, *HRP*, and *IRMA-8* are reported in Tables (3.1), (3.2), and (3.3) respectively. The degree of redundancy for these robots is $\mu = 2$, yielding 28 possible combinations of length- and force-controlled cables [43]. All possible combinations were analyzed to assess how the shape and volume of the *WFEI* workspace vary with different control combinations and robot architectures.

In the following, only a part of the full analysis is presented to focus on the most relevant features and results.

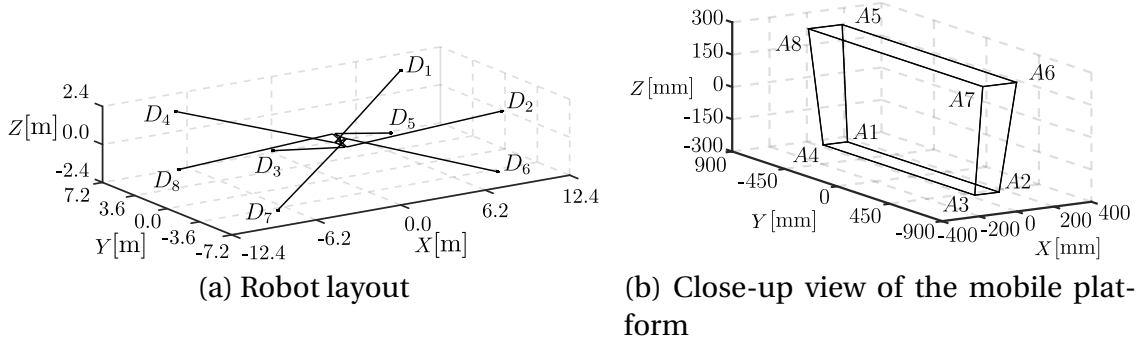


Figure 3.1: *IPanema 3 OCDPR*

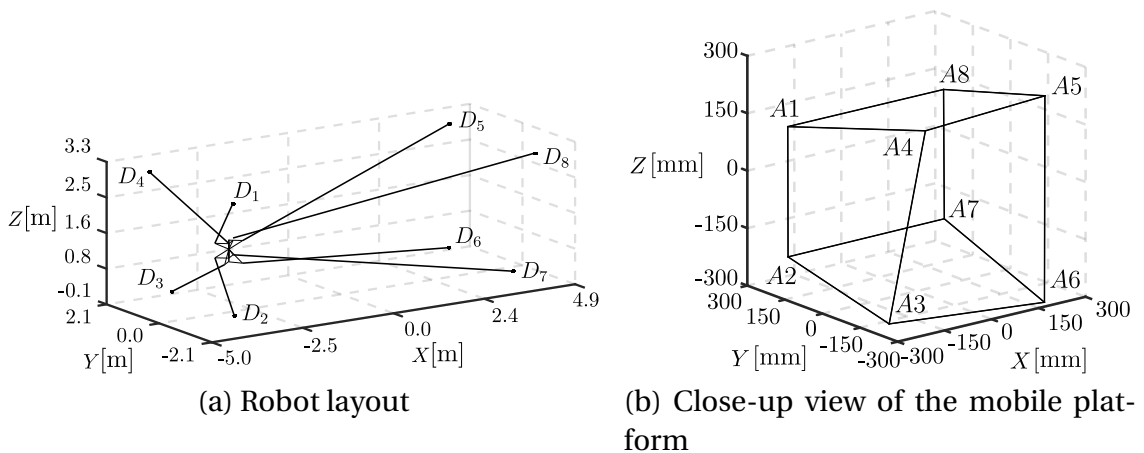
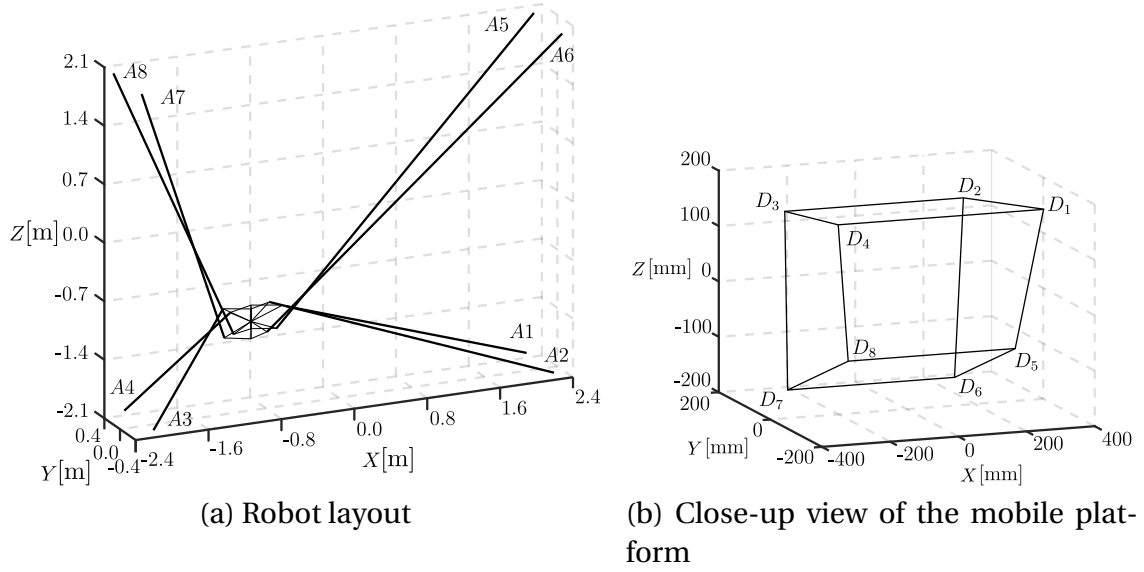


Figure 3.2: *HRP OCDPR*


 Figure 3.3: IRMA-8 deployable *OCDPR*

i	1	2	3	4	5	6	7	8
$\mathbf{d}_i$ [m]	$\begin{bmatrix} 8.500 \\ 6.000 \\ 2.250 \end{bmatrix}$	$\begin{bmatrix} 8.500 \\ -6.000 \\ 2.250 \end{bmatrix}$	$\begin{bmatrix} -8.500 \\ -6.000 \\ 2.250 \end{bmatrix}$	$\begin{bmatrix} -8.500 \\ 6.000 \\ 2.250 \end{bmatrix}$	$\begin{bmatrix} 8.500 \\ 6.000 \\ -2.250 \end{bmatrix}$	$\begin{bmatrix} 8.500 \\ -6.000 \\ -2.250 \end{bmatrix}$	$\begin{bmatrix} -8.500 \\ -6.000 \\ -2.250 \end{bmatrix}$	$\begin{bmatrix} -8.500 \\ 6.000 \\ -2.250 \end{bmatrix}$
$\mathbf{a}'_i$ [m]	$\begin{bmatrix} 0.113 \\ 0.750 \\ -0.250 \end{bmatrix}$	$\begin{bmatrix} 0.113 \\ -0.750 \\ -0.250 \end{bmatrix}$	$\begin{bmatrix} -0.113 \\ -0.750 \\ -0.250 \end{bmatrix}$	$\begin{bmatrix} -0.113 \\ 0.750 \\ -0.250 \end{bmatrix}$	$\begin{bmatrix} 0.113 \\ 0.750 \\ 0.250 \end{bmatrix}$	$\begin{bmatrix} 0.113 \\ -0.750 \\ 0.250 \end{bmatrix}$	$\begin{bmatrix} -0.113 \\ -0.750 \\ 0.250 \end{bmatrix}$	$\begin{bmatrix} -0.113 \\ 0.750 \\ 0.250 \end{bmatrix}$

 Table 3.1: Geometrical parameters of the IPAnema 3 *OCDPR*

i	1	2	3	4	5	6	7	8
$\mathbf{d}_i$ [m]	$\begin{bmatrix} -4.195 \\ -1.776 \\ 3.014 \end{bmatrix}$	$\begin{bmatrix} -3.853 \\ -1.329 \\ 0.243 \end{bmatrix}$	$\begin{bmatrix} -3.783 \\ 1.363 \\ 0.244 \end{bmatrix}$	$\begin{bmatrix} -4.097 \\ 1.809 \\ 3.015 \end{bmatrix}$	$\begin{bmatrix} 4.102 \\ 1.782 \\ 3.021 \end{bmatrix}$	$\begin{bmatrix} 3.787 \\ 1.332 \\ 0.248 \end{bmatrix}$	$\begin{bmatrix} 3.772 \\ -1.340 \\ 0.249 \end{bmatrix}$	$\begin{bmatrix} 4.083 \\ -1.789 \\ 3.019 \end{bmatrix}$
$\mathbf{a}'_i$ [m]	$\begin{bmatrix} -0.249 \\ 0.202 \\ 0.129 \end{bmatrix}$	$\begin{bmatrix} -0.248 \\ 0.202 \\ -0.211 \end{bmatrix}$	$\begin{bmatrix} -0.249 \\ -0.202 \\ -0.288 \end{bmatrix}$	$\begin{bmatrix} -0.191 \\ -0.274 \\ 0.223 \end{bmatrix}$	$\begin{bmatrix} 0.248 \\ -0.202 \\ 0.211 \end{bmatrix}$	$\begin{bmatrix} 0.189 \\ -0.275 \\ -0.299 \end{bmatrix}$	$\begin{bmatrix} 0.250 \\ 0.202 \\ -0.209 \end{bmatrix}$	$\begin{bmatrix} 0.248 \\ 0.202 \\ 0.129 \end{bmatrix}$

 Table 3.2: Geometrical parameters of the HRP *OCDPR*

i	1	2	3	4	5	6	7	8
$\mathbf{a}_i$ [m]	$\begin{bmatrix} 2.294 \\ 0.3325 \\ -1.898 \end{bmatrix}$	$\begin{bmatrix} 2.284 \\ -0.377 \\ -1.898 \end{bmatrix}$	$\begin{bmatrix} -2.284 \\ -0.377 \\ -1.898 \end{bmatrix}$	$\begin{bmatrix} -2.294 \\ 0.332 \\ -1.898 \end{bmatrix}$	$\begin{bmatrix} 2.398 \\ 0.377 \\ 1.794 \end{bmatrix}$	$\begin{bmatrix} 2.398 \\ -0.332 \\ 1.784 \end{bmatrix}$	$\begin{bmatrix} -2.398 \\ -0.332 \\ 1.784 \end{bmatrix}$	$\begin{bmatrix} -2.398 \\ 0.377 \\ 1.794 \end{bmatrix}$
$\mathbf{d}'_i$ [m]	$\begin{bmatrix} 0.302 \\ -0.134 \\ 0.180 \end{bmatrix}$	$\begin{bmatrix} 0.262 \\ 0.127 \\ 0.138 \end{bmatrix}$	$\begin{bmatrix} -0.262 \\ 0.127 \\ 0.138 \end{bmatrix}$	$\begin{bmatrix} -0.302 \\ -0.134 \\ 0.180 \end{bmatrix}$	$\begin{bmatrix} 0.245 \\ -0.097 \\ -0.078 \end{bmatrix}$	$\begin{bmatrix} 0.245 \\ 0.138 \\ -0.188 \end{bmatrix}$	$\begin{bmatrix} -0.245 \\ 0.138 \\ -0.188 \end{bmatrix}$	$\begin{bmatrix} -0.245 \\ -0.097 \\ -0.078 \end{bmatrix}$

 Table 3.3: Geometrical parameters of the IRMA-8 deployable *OCDPR*

The values of $\delta\tau$ and δl should be selected based on multiple factors, including sensor uncertainties, dynamic response, external disturbances, and unmodeled effects. Practical engineering values may fall within the ranges $\delta\tau = 1\% \div 5\% \tau_{max}$ and

$\delta l = 0.1\% \div 1\% l_{max}$, where τ_{max} represents the maximum cable tension and l_{max} the maximum cable length of the prototype under analysis.

The following parameters were used as an example:

- $\tau_{min} = 50$ N and $\tau_{max} = 2000$ N as tension bounds;
- $\delta\tau = 20$ N as loadcell uncertainty, assumed to be 1% of τ_{max} ;
- $\delta l = 10$ mm, as cable-length error, assumed to be 1% of the maximum cable length of the robots under consideration;
- $\sigma_{\infty, max}^{p, l_d} = 5$ mm/mm and $\sigma_{\infty, max}^{\epsilon, l_d} = 0.5$ °/mm³ as kinematic sensitivity limits, set as minimum precision requirements for both robots (corresponding to $\|\mathbf{dp}_{max}\|_{\infty} = 50$ mm and $\|\mathbf{d\epsilon}_{max}\|_{\infty} = 5^\circ$ according to Eq.(3.6) and the previous δl value).

The installation volume was discretized into 8000 points, with the x , y , and z installation dimensions divided into 19 equidistant segments. The *EE* orientation was held constant at $\mathbf{e}^T = [0 \ 0 \ 0]$ rad. The inertial parameters were taken from [43] for the IPAnema 3, with the platform having a mass of $m = 50$ Kg and a center of mass located at $\mathbf{s}'^T = [0 \ 0 \ 0]$ m, and from [45] for the HRP Cable Robot, with the platform having a mass of $m = 23$ Kg and $\mathbf{s}'^T = [0 \ 0 \ 0]$ m.

Figures from (3.4) to (3.9) show the results for two force- and length-controlled cable combinations for each *OCDPR* architecture. In these figures, cake diagrams illustrate the ratio between the installation space and various *WFW* sub-regions. The Not-Wrench-Feasible (*NWF*) workspace is not shaded. It is evident that the configuration and dimensions of the *WFEI* workspace change significantly when different combinations of length- and force-controlled cables are employed.

Regarding IPAnema 3, selecting cables 4 and 6 for force control nearly doubles the volume of the *WFEI* workspace compared to using cables 1 and 2. This substantial increase is due to differences in the kinematic accuracy achieved in each configuration. Specifically, the Kinematic Sensitive (*KS*) region in Figure (3.4) is more than double than in Figure (3.5), highlighting that force-controlling cables 1 and 2 does not meet the desired kinematic accuracy as effectively as controlling cables 4 and 6. Meanwhile, the Tension Sensitive (*TS*) workspace remains relatively stable between the two configurations, indicating that this region is less affected by the selection of force-controlled cables.

Figures (3.6) and (3.7) show results for two force- and length-control configurations of the HRP - Cable Robot. In the first configuration (Figure 3.6), cables 2 and 8 are force-controlled, while in the second one (Figure 3.7), cables 2 and 7 are force-controlled. For the HRP robot, force-controlling cables 2 and 7 is the more advantageous configuration. In Figure (3.6a), the *TS* workspace occupies 12% more of the volume than in Figure (3.7a). This difference suggests that using cables 2 and 7 for force control minimizes the *TS* region. Another critical observation in Figure (3.6a) is the division of the *WFEI* workspace into two distinct regions. To improve the visualization of these *WFEI* isolated regions, cross-sectional views at $y = -1.04$ m, $y = -0.40$ m, $y = 0.25$ m, and $y = 0.90$ m are provided in Fig. 3.10. To transition between these regions, the *EE* must pass through a *TS* region where slackening of cables may occur due to tension

³The limit on the maximum absolute value of an orientation variation has a physical meaning only assuming small rotations, as pointed out in [96].

control errors, potentially leading to loss of positioning accuracy. Hence, a configuration with cables 2 and 8 under force control is less desirable. Conversely, as seen in Figure (3.7a), cables 2 and 7 maintain a continuous *WFEI* region, avoiding such issues and providing a safer and more robust workspace for task execution.

Finally, the workspace analysis for the IRMA-8 deployable prototype is shown in Figures (3.8) and (3.9). Similar to the previous architectures, the choice of force-controlled cables significantly impacts the distribution of sub-regions within the installation space. When cables 1 and 2 are selected for force control (Figure 3.8), the *WFEI* workspace is notably restricted, accounting for only 10% of the analyzed volume. In this configuration, a small portion of the workspace is categorized as Tension Sensitive at 10%, and Kinematic Sensitive at 28%, suggesting that this combination is particularly susceptible to both kinematic inaccuracies. In contrast, force-controlling cables 2 and 3 (Figure 3.9) represents the optimal combination for the IRMA-8. This selection expands the *WFEI* workspace to 22% of the total volume. This improvement is primarily driven by a sharp reduction in kinematic sensitivity; the *KS* region drops from 28% to just 17%. While the *TS* region has a slight reduction to 9% in this configuration, the overall gain in the high-precision *WFEI* zone makes this combination the most effective for ensuring reliable robot performance across the largest possible area.

Figures (3.5), (3.7), and (3.9) represent the overall optimal cable-control combinations that yield the maximum *WFEI* workspace volume for each robot. Selecting cables 4 and 6 for the IPAnema 3, cables 2 and 7 for the HRP robot, and cables 2 and 3 for the IRMA-8 maximize the *WFEI* workspace relative to installation space. Assuming that all robots use identical sensors and actuators and thus share similar control errors and kinematic sensitivity limits, the IPAnema 3, despite having a smaller *WFW* to installation-space ratio compared to the HRP Cable Robot, achieves a larger *WFEIW* to installation-space ratio when a *HC* is applied.

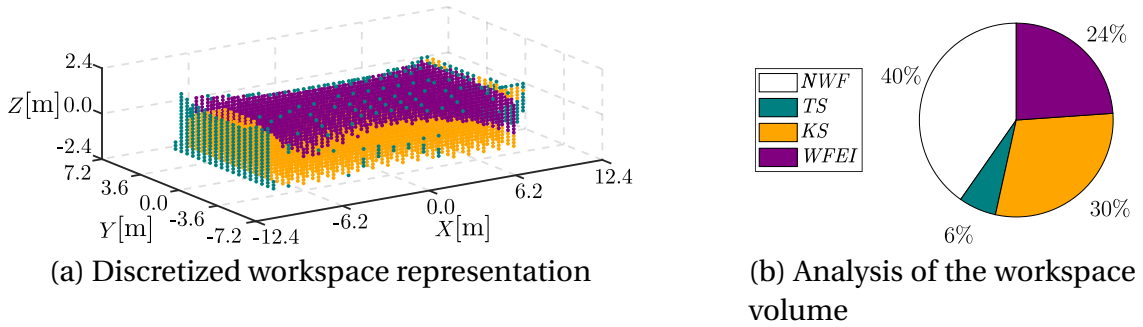


Figure 3.4: IPAnema 3 Workspace with 1-2 as force-controlled cables

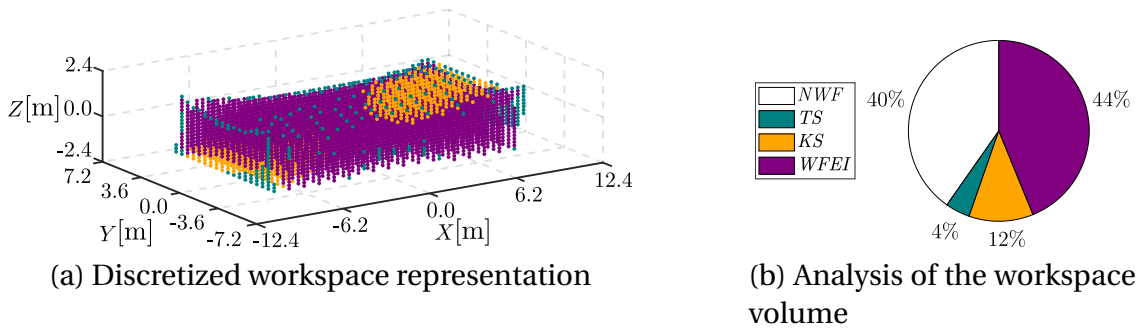


Figure 3.5: IPAnema 3 Workspace with 4-6 as force-controlled cables

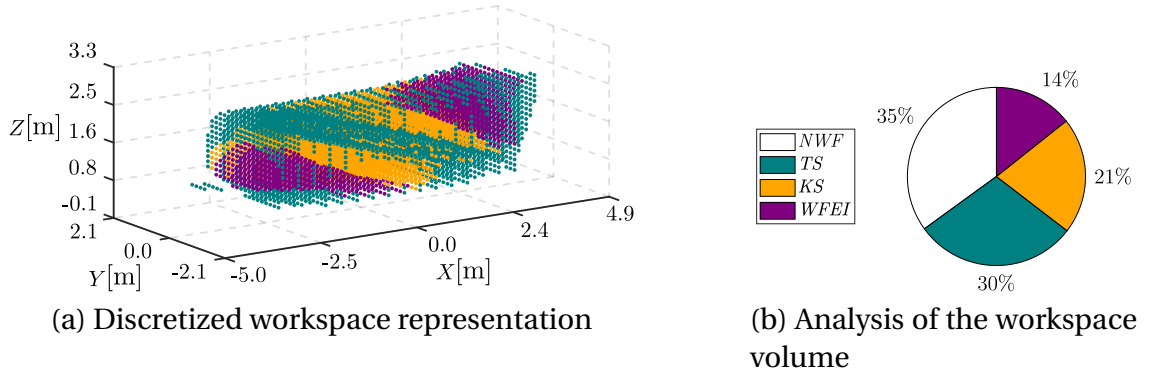


Figure 3.6: HRP Workspace with 2-8 as force-controlled cables

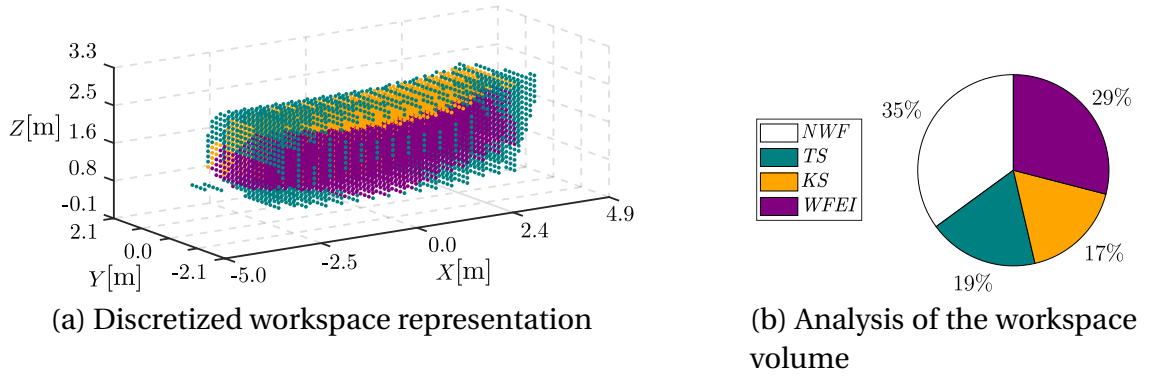


Figure 3.7: HRP Workspace with 2-7 as force-controlled cables

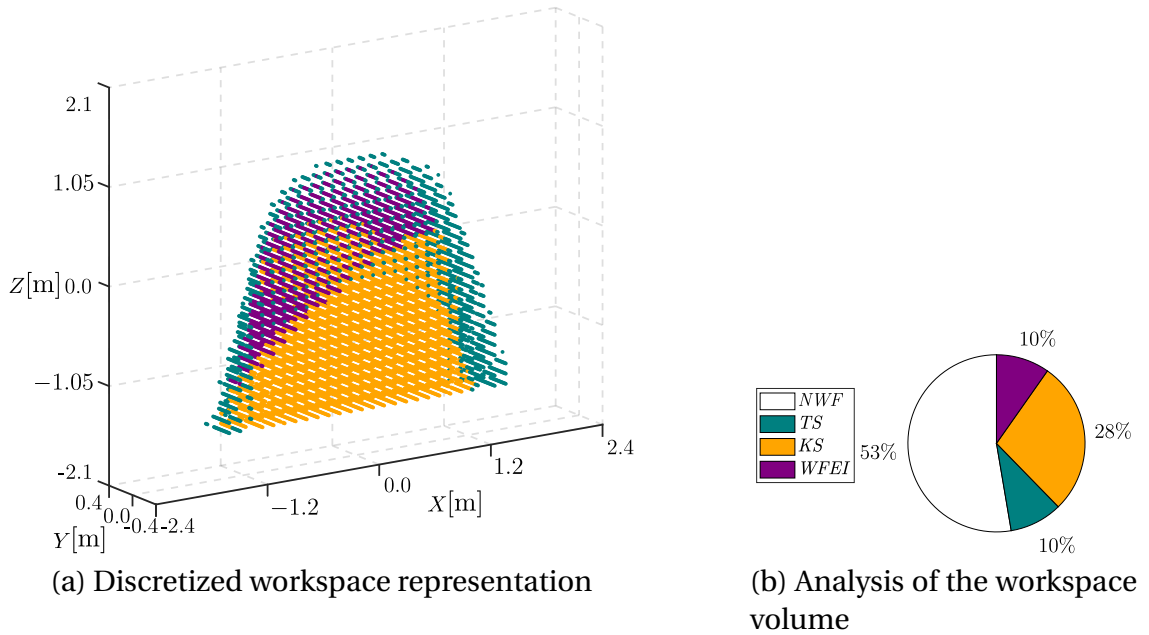


Figure 3.8: IRMA-8 Workspace with 1-2 as force-controlled cables

3.3 Design Optimization algorithm

The workspace analysis presented in Section 3.2 can be used in the design phase of the robot *EE* to maximize the usable workspace for a given installation space. Since the force- and length-controlled cable combination c strongly influences the *WFEI* workspace, it can be included in the optimization problem as an unknown variable to be

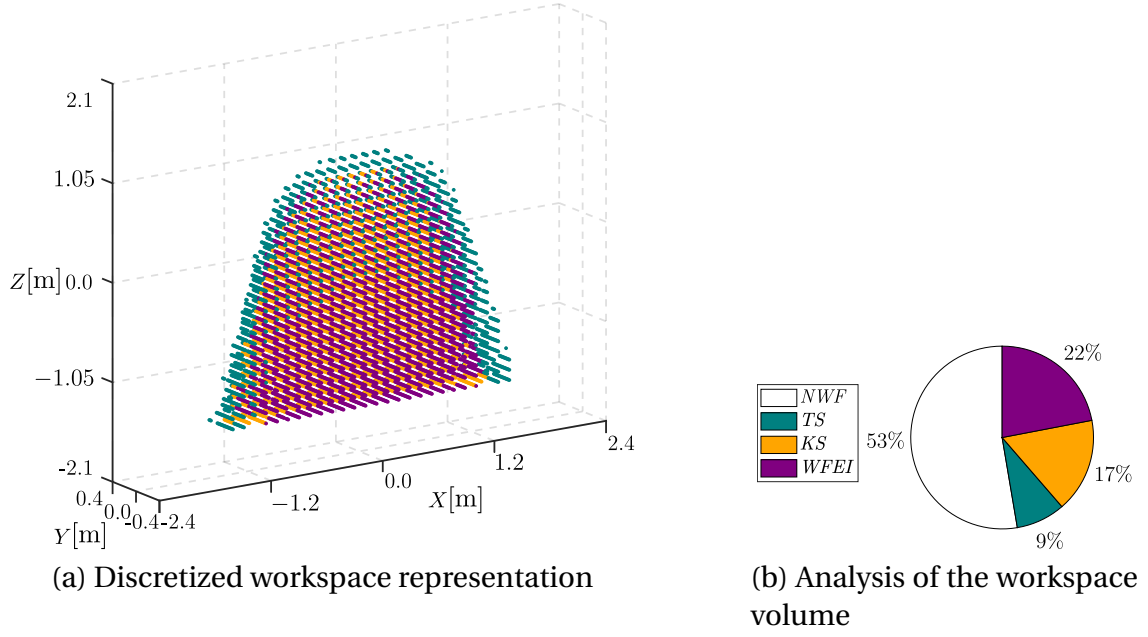
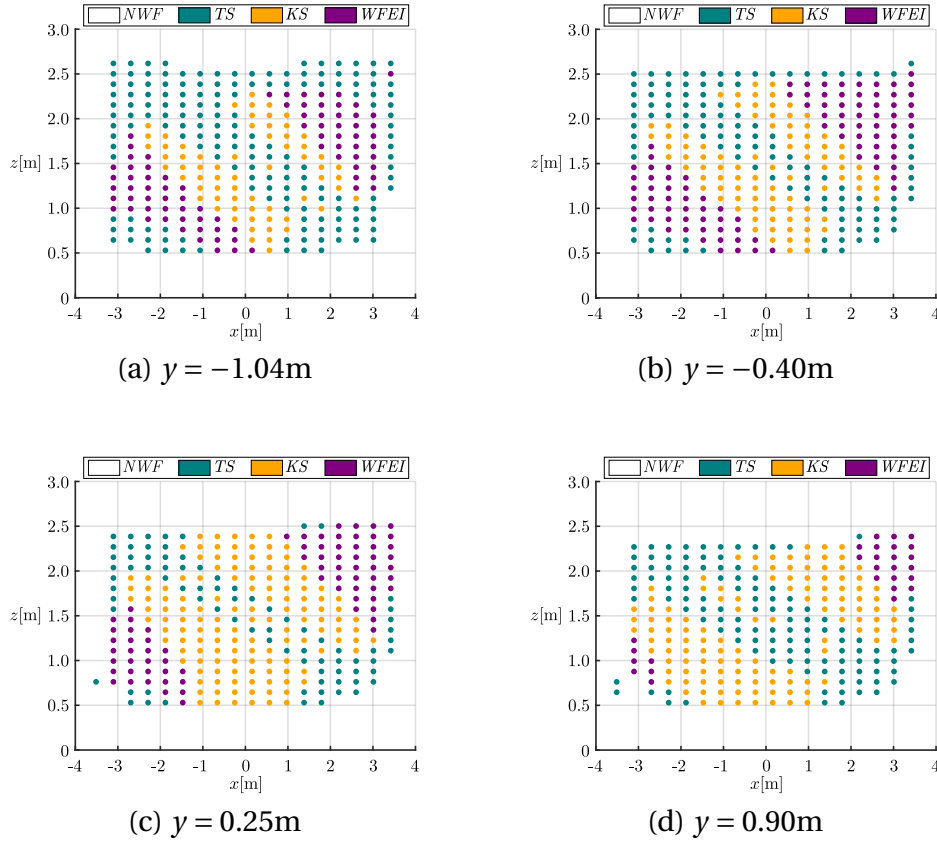


Figure 3.9: IRMA-8 Workspace with 2-3 as force-controlled cables


 Figure 3.10: HRP Workspace with 2-8 as force-controlled cables. Cross sections in x-z planes at $y = -1.04\text{m}$, $y = -0.40\text{m}$, $y = 0.25\text{m}$, $y = 0.90\text{m}$.

determined. Therefore, the unknown variables of the design optimization problem are:

$$\begin{cases} \mathbf{a}'_i \in \mathbb{R}^3, & \text{with } i = 1, \dots, n \\ c \in [1, \dots, 28], & \text{with } c \in \mathbb{N}, \end{cases} \quad (3.39)$$

The cost function $f(\mathbf{X}, c)$ is the ratio of the *WFEIW* volume V_{WFEIW} to the installation space volume V_{inst} :

$$f(\mathbf{X}, c) = \frac{V_{WFEIW}(\mathbf{X}, c)}{V_{inst}} \quad (3.40)$$

where $\mathbf{X} = [\mathbf{a}'_1, \dots, \mathbf{a}'_n]^T$, and V_{inst} is defined by the convex interpolation of the frame anchor points B_i . While V_{WFEIW} could serve directly as the cost function, using the ratio in Eq. (3.40) ensures a physically meaningful metric that remains independent of the robot specific architecture. Maximizing the objective function (3.40) allows the manipulation of the robot in a bigger region in which the accuracy performance requirements are met, and wrench feasibility is ensured even in the presence of control errors. Considering all variables in Eq. (3.39) in a single optimization results in a combined continuous⁴ and discrete problem that cannot be solved by a classical gradient-based optimizer, and it is challenging even for population-based optimizers (such as genetic algorithms). For this reason, a nested optimization algorithm was developed that divides optimization into two steps. The algorithm structure is represented in Fig. (3.11), and its steps are detailed as follows:

1. the first step consists in determining the best combination of length- and force-controlled cables c_{old} , computing the *WFEI* workspace volume for all discrete values of c and maintaining the geometric parameters $\mathbf{X}_0$ of the initial architecture constant;
2. secondly, imposing boundary constraints on the values of $\mathbf{X}$, the following optimization problem is solved by using a gradient-based solver:

$$\max_{\mathbf{X}} f(\mathbf{X}, c_{old}) \quad \text{s.t.} \quad \mathbf{X}_{lb} \leq \mathbf{X} \leq \mathbf{X}_{ub} \quad (3.41)$$

where the boundary constraints $\mathbf{X}_{lb} = [\mathbf{a}'_{1,min}, \dots, \mathbf{a}'_{n,min}]^T$ and $\mathbf{X}_{ub} = [\mathbf{a}'_{1,max}, \dots, \mathbf{a}'_{n,max}]^T$ are chosen as the robot footprint limits and are reported in Table 3.4 for the IPAnema 3. The solution of problem (3.41) is stored in the variable $\mathbf{X}_{new}$;

3. the third step consists in repeating the exhaustive search of the best value of c with the new platform architecture $\mathbf{X}_{new}$. If the new combination c_{new} is equal to the cable combination found with the older platform geometry c_{old} , the algorithm terminates; otherwise, the c value is updated, and the algorithm returns to the second step.

While alternatives such as a mixed-integer genetic algorithm *GA* or iterative optimization of c after each gradient-based step could be considered, the nested approach provides a more tailored solution to the problem's characteristics. A mixed-integer *GA* allows simultaneous optimization of continuous variables ($\mathbf{X}$) and discrete variables (c), which can explore large search spaces. However, the exhaustive computation of the fitness function during *GA* iterations would significantly increase the computational overhead, especially since evaluating the workspace volume is resource-intensive.

Alternatively, iterative optimization of c at every gradient-based step would partially address the coupling of variables but could lead to inefficient convergence due

⁴in this study, the optimization variables (representing geometric parameters) are considered as continuous values.

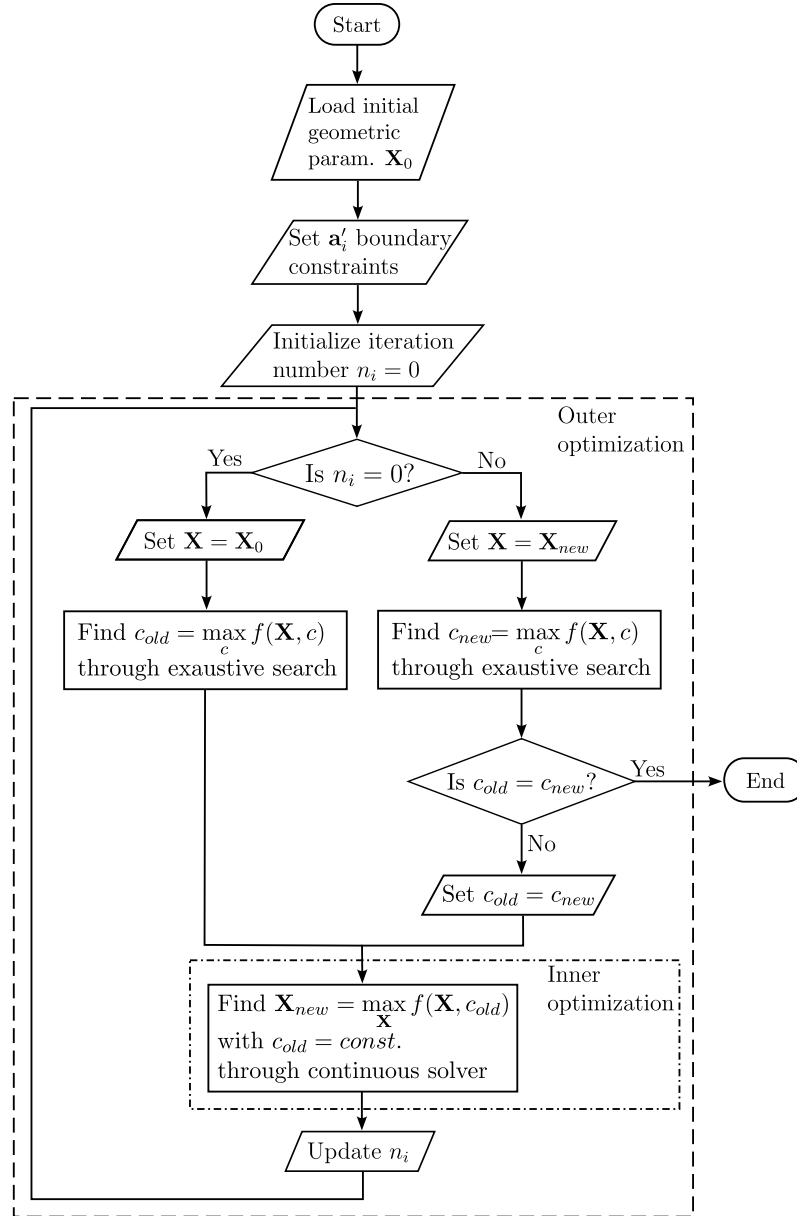


Figure 3.11: Nested optimization algorithm

i	1	2	3	4	5	6	7	8
$\mathbf{a}'_{i,min}$ [m]	$\begin{bmatrix} 0.03 \\ 0.32 \\ -0.40 \end{bmatrix}$	$\begin{bmatrix} 0.03 \\ -0.97 \\ -0.40 \end{bmatrix}$	$\begin{bmatrix} -0.10 \\ -0.97 \\ -0.40 \end{bmatrix}$	$\begin{bmatrix} -0.10 \\ 0.32 \\ -0.40 \end{bmatrix}$	$\begin{bmatrix} 0.05 \\ 0.37 \\ 0.13 \end{bmatrix}$	$\begin{bmatrix} 0.05 \\ -1.12 \\ 0.13 \end{bmatrix}$	$\begin{bmatrix} -0.13 \\ -1.12 \\ 0.13 \end{bmatrix}$	$\begin{bmatrix} -0.13 \\ 0.37 \\ 0.13 \end{bmatrix}$
$\mathbf{a}'_{i,max}$ [m]	$\begin{bmatrix} 0.09 \\ 0.97 \\ -0.13 \end{bmatrix}$	$\begin{bmatrix} 0.09 \\ -0.33 \\ -0.13 \end{bmatrix}$	$\begin{bmatrix} -0.04 \\ -0.33 \\ -0.13 \end{bmatrix}$	$\begin{bmatrix} -0.04 \\ 0.97 \\ -0.13 \end{bmatrix}$	$\begin{bmatrix} 0.14 \\ 1.12 \\ 0.40 \end{bmatrix}$	$\begin{bmatrix} 0.14 \\ -0.37 \\ 0.40 \end{bmatrix}$	$\begin{bmatrix} -0.04 \\ -0.37 \\ 0.40 \end{bmatrix}$	$\begin{bmatrix} -0.04 \\ 1.12 \\ 0.40 \end{bmatrix}$

Table 3.4: Boundary constraints on the IPAnema 3 platform geometry

to repeated recalculations of c without significant progress. In contrast, the proposed nested algorithm leverages the manageable size of the discrete search space for c , allowing for a deterministic and computationally efficient exhaustive search in the discrete domain. By decoupling c and $\mathbf{X}$, the algorithm may be more robust and aligned with the nature of the cost function for small perturbations in $\mathbf{X}$. Thus, the nested al-

gorithm balances computational efficiency and optimization accuracy.

As an application example of the design optimization, the algorithm was applied to the *IPanema 3* architecture since it demonstrated a greater *WFEIW*-to-installation space ratio than the *HRP* when considering the best control scheme in the analysis of Section 3.2.1. For this reason, the *IPanema 3* design was considered a better starting point for the nested optimization algorithm. Additionally, to evaluate the impact of force control errors on the design and control optimization outcomes, the *IPanema 3* architecture was optimized both in the case $\delta\tau = 1\%\tau_{max}$ and in the worst-case $\delta\tau = 5\%\tau_{max}$ (see Sec. 3.2.2). The MATLAB function *fmincon*, with an interior-point method, was used to solve the nonlinear optimization problem of the second step. The volume cost function $f(\mathbf{X}, c)$ was computed using the MATLAB *boundary* function, which creates an alpha shape [99] from the point cloud belonging to the *WFEI* workspace.

The initial platform architecture is the one represented in Fig. 3.1b and detailed in Tab. 3.1. Using $\delta\tau = 1\%\tau_{max}$, the combination of length- and force-controlled cables that maximizes $f(\mathbf{X}, c)$ with the starting geometry is $c = 20$, which corresponds to cables 4 and 6 to be force-controlled, as highlighted in Section 3.2.2. After three iterations, the optimization algorithm converged into the platform geometry shown in Figure 3.12 and detailed in Tab. 3.5, in which the resulting optimal combination of force-controlled cables is 2-8. The corresponding is shown in Fig. 3.13, which, compared to the one obtained with the initial geometry and control-scheme represented in Fig. 3.5, is increased by 17% of the installation space.

With $\delta\tau = 5\%\tau_{max}$, the initial optimal force-controlled cable combination changes to $c = 6$, corresponding to cables 1 and 7. As expected, the increased control error leads to a slightly reduced initial *WFEIW*, as shown in Fig. 3.14a, compared to the previous case in Fig. 3.5. After applying the design and control optimization, the optimal force-controlled cable combination remains $c = 6$, and the final *WFEIW* increases by 12% (Fig. 3.15b). The design optimization results are represented in Fig. 3.16 and detailed in Tab. 3.6. Therefore, in this case, higher force errors influence the optimal selection of length- and force-controlled cables and result in a lower workspace improvement after optimization.

The design optimization algorithm was executed on a PC equipped with a 13th-generation Intel i7 processor and 32 GB of RAM. The total computation time from *Start* to *End* of the nested optimization algorithm represented in Fig. 3.11 was 34.6 hours, with an average of 14.9 minutes per single workspace computation. To reduce computation time, the optimization workload was distributed across 14 cores.

i	$^P \mathbf{a}_i'$ [m]
1	$[0.0359 \ 0.7630 \ -0.3648]^T$
2	$[0.0588 \ -0.5818 \ -0.3949]^T$
3	$[-0.0765 \ -0.6325 \ -0.3881]^T$
4	$[-0.0641 \ 0.8644 \ -0.3951]^T$
5	$[0.1283 \ 1.0930 \ 0.2202]^T$
6	$[0.1335 \ -1.1130 \ 0.3969]^T$
7	$[-0.0987 \ -0.9833 \ 0.2526]^T$
8	$[-0.1237 \ 0.9568 \ 0.2591]^T$

Table 3.5: Optimized attachment points of the IPAnema 3 platform with $\delta\tau = 1\%\tau_{max}$

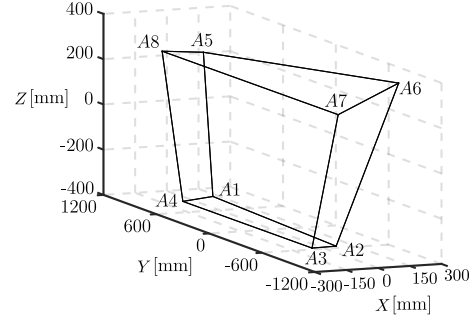
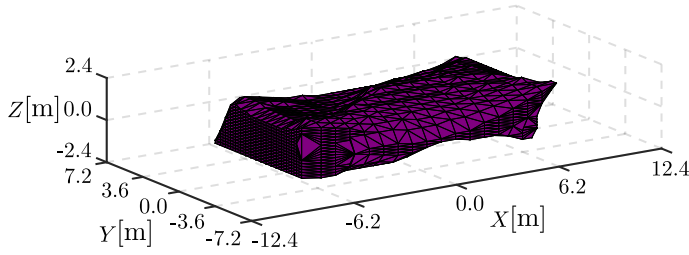
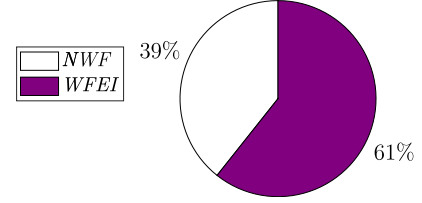


Figure 3.12: Optimized geometry of the IPAnema 3 platform with $\delta\tau = 1\%\tau_{max}$

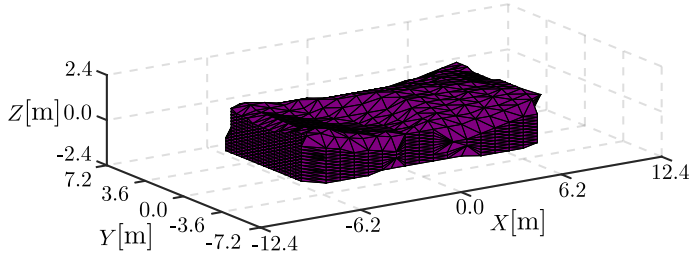


(a) IPAnema 3 WFEI workspace after platform design optimization, with 2-8 as force-controlled cables

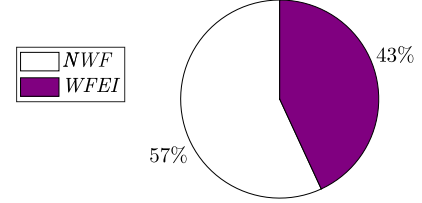


(b) Ratio between the WFEI workspace and the installation space volumes

Figure 3.13: IPAnema 3 platform design optimization with $\delta\tau = 1\%\tau_{max}$

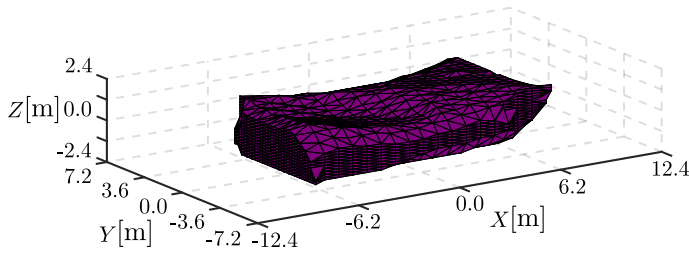


(a) IPAnema 3 WFEI workspace with 1-7 as force-controlled cables

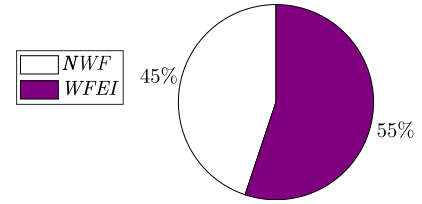


(b) Ratio between the WFEI workspace and the installation space volumes

Figure 3.14: IPAnema 3 WFEIW before optimization, with $\delta\tau = 5\%\tau_{max}$



(a) IPAnema 3 WFEI workspace after platform design optimization, with 1-7 as force-controlled cables



(b) Ratio between the WFEI workspace and the installation space volumes

Figure 3.15: IPAnema 3 platform design optimization with $\delta\tau = 5\%\tau_{max}$

i	${}^P\mathbf{a}'_i$ [m]
1	$[0.0333 \quad 0.5142 \quad -0.3924]^T$
2	$[0.0361 \quad -0.4312 \quad -0.3885]^T$
3	$[-0.0535 \quad -0.7211 \quad -0.3917]^T$
4	$[-0.0477 \quad 0.5932 \quad -0.3899]^T$
5	$[0.1353 \quad 1.7683 \quad 0.3938]^T$
6	$[0.1301 \quad -0.9699 \quad 0.1949]^T$
7	$[-0.0987 \quad -0.6244 \quad 0.1509]^T$
8	$[-0.0604 \quad 0.6943 \quad 0.2464]^T$

Table 3.6: Optimized attachment points of the IPAnema 3 platform with $\delta\tau = 5\%\tau_{max}$

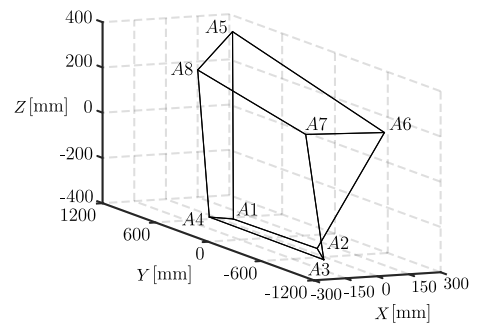


Figure 3.16: Optimized geometry of the IPAnema 3 platform with $\delta\tau = 5\%\tau_{max}$

Chapter 4

Initial-Pose Self-Calibration methods for CDPRs

This chapter presents the core research of this thesis, focusing on the problem of initial-pose self-calibration¹. The operational accuracy of a *CDPR* in real-world scenarios is strictly bounded by the precision of its state estimation. As highlighted in the introductory section 1, *CDPRs* often rely on relative internal sensors, such as incremental rotary encoders, which only measure variations from an unknown starting configuration. Consequently, at every system startup, the absolute reference values of these sensors, and by extension the initial and subsequent poses of the *EE*, are unknown [56–59, 62, 100, 101].

The challenge of determining these reference values without relying on expensive or impractical external measurement systems (e.g., laser trackers or motion capture [55, 102]) defines the field of self-calibration. This procedure is particularly critical for large-scale and reconfigurable systems, where geometric parameters may shift after installation or relocation [103]. By leveraging the robot's onboard sensors, self-calibration provides a cost-effective and rapid setup procedure, essential for the usage of *CDPRs* in industrial and on-field applications.

The research presented in this chapter addresses the limitations of existing state-of-the-art methods, which suffer from long data-acquisition times [62], or lack a systematic framework for self-calibration parameter selection. The contributions detailed hereafter span from the development of rapid, automated data-acquisition strategies for underactuated architectures to a comprehensive Monte Carlo analysis of the algorithmic parameters with the objective of ensuring optimal convergence and accuracy of the solution. Finally, this chapter introduces novel online self-calibration techniques, representing the latest advancement in acquiring the immediate operational readiness and precision of *CDPRs*.

The following sections are organized as follows.

Section 4.1 provides a generic formulation of the initial-pose self-calibration problem. It defines the mathematical structure of the Nonlinear Weighted Least-Squares (*NWLS*) optimization, where the residuals between measured and modeled quantities are minimized to estimate the unknown initial state.

Section 4.2 focuses on the practical challenges of data acquisition for *UACDPRs*. Motion control of these systems, when a rough estimate of the initial pose is available,

¹The content of this chapter is partially based on [52, 60, 61] submitted to *Robotica*, *Advances in Service and Industrial Robotics* and *Advances in Italian Mechanism Science*.

doesn't address the problem of residual oscillations given by the free motion, and traditional methods require long waiting times for passive damping. This section presents novel rapid manual and fully automated techniques that utilize low-level tension control and sensor fusion to reduce the time required for data acquisition and the overall calibration.

Section 4.3 offers a systematic evaluation of the key parameters influencing calibration performance. Building on a preliminary comparative study of deployable architectures and an extensive Monte Carlo simulation campaign, this section analyzes the impact of the number of measurement poses, the choice of the sensor set (including motor encoders, swivel encoders and *AHRS*), and the robustness of various non-linear solvers (such as Levenberg-Marquardt and Trust-Region Newton methods using the real Hessian).

Section 4.4 introduces novel online initial-pose self-calibration techniques. By utilizing Recursive Least Squares (*RLS*) and Kalman filtering, these methods move beyond traditional offline batch processing and online nested approaches [59], allowing for a more streamlined and adaptive startup procedure.

4.1 The initial-pose self-calibration problem

In this section, a generic formulation of the initial-pose self-calibration problem is established. All geometric parameters of the *CDPR*, sensor calibration parameters, and inertial quantities are assumed to be known and constant throughout the procedure.

Let $\zeta_0 \in \mathbb{R}^6$ denote the unknown initial *EE* pose, and let ζ_k ($k = 1, \dots, N$) denote a sequence of poses attained by the *EE* after the initial pose, and where sensor data is recorded. Since no external sensing is available, the poses ζ_k cannot be directly measured and must be estimated together with the initial pose ζ_0 as a byproduct of the self-calibration. To reduce the impact of measurement noise and increase observability, the number of measurement poses N has to be sufficiently large to obtain an overdetermined system of equations [61]. Let $\mathbf{y}_k(\zeta_k) \in \mathbb{R}^m$ denote the array of modelled values of the m physical quantities associated with the sensor measurements at pose ζ_k (e.g., nominal cable lengths, swivel angles, and orientation components). Measured quantities are denoted by the superscript $(\cdot)^*$. Measurement noise is represented by $\mathbf{v}_y \in \mathbb{R}^m$ and is assumed to be independent of the *EE* pose, zero-mean, and Gaussian, i.e., $\mathbf{v}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_v)$, with $\mathbf{Q}_v \in \mathbb{R}^{m \times m}$. The same noise covariance $\mathbf{Q}_v$ is assumed for all measurement poses. Each sensor output belongs either to the class of *relative measurements* (e.g., cable-length variations, swivel-angle variations, and yaw variations) or to the class of *absolute measurements* (e.g., roll and pitch with respect to earth gravity, aligned with the fixed robot frame). For the j -th sensor measurement, the residual at pose k is defined as

$$r_{j,k} = y_{j,k}(\zeta_k) - y_{j,0}(\zeta_0) - \Delta y_{j,k}^* - \mathbf{v}_{y_j}, \quad \text{for relative measurements,} \quad (4.1)$$

$$r_{j,k} = y_{k,j}(\zeta_k) - y_{j,k}^* - \mathbf{v}_{y_j}, \quad \text{for absolute measurements,} \quad (4.2)$$

where $\Delta y_{j,k}^*$ denotes the measured variation of the j -th quantity between pose ζ_k and the initial pose ζ_0 . The modeled quantities $y_{j,k}(\zeta_k)$ depend on the *EE* pose through the kinematic relations introduced in Sections 2.1 and 2.3. For relative measurements, the initial reference values are unknown and appear in the residuals through $y_{j,0}(\zeta_0)$. Under the adopted kinematic model, these reference values are uniquely determined by

the initial pose, i.e., $y_0 = y(\zeta_0)$, and therefore no additional parameters are introduced. As a result, estimating ζ_0 is equivalent to estimating the initial reference values of all relative measurements.

Stacking all measurement poses, the vector of unknowns and the global residual vector are defined as

$$\mathbf{x} = \begin{bmatrix} \zeta_0 \\ \zeta_1 \\ \vdots \\ \zeta_N \end{bmatrix} \in \mathbb{R}^{6(N+1)}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_N \end{bmatrix} \in \mathbb{R}^{mN}, \quad (4.3)$$

where each block $\mathbf{r}_k \in \mathbb{R}^m$ is defined component-wise according to Eq. (4.1) or Eq. (4.2).

If the system is overdetermined, i.e., if $mN > 6(N+1)$, the initial-pose self-calibration can be formulated as a *NWLS* problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} f(\mathbf{x}), \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{r}^T \mathbf{W} \mathbf{r}. \quad (4.4)$$

The weighting matrix $\mathbf{W} \in \mathbb{R}^{mN \times mN}$ is chosen as the inverse of the covariance matrix of the residual vector [59]:

$$\mathbf{W} = (\text{Cov}(\mathbf{r}))^{-1} = \mathbf{I}_{N \times N} \otimes (\mathbf{J}_v \mathbf{Q}_v \mathbf{J}_v^T)^{-1}, \quad (4.5)$$

where $\otimes$ denotes the Kronecker product, $\mathbf{I}_{N \times N} \in \mathbb{R}^{N \times N}$ is the identity matrix, and $\mathbf{J}_v = \partial \mathbf{r}_k / \partial \mathbf{v}_y \in \mathbb{R}^{m \times m}$. Since the dependence of the residuals on the measurement noise is linear in Eqs. (4.1)–(4.2), we have $\mathbf{J}_v \mathbf{Q}_v \mathbf{J}_v^T = \mathbf{Q}_v$, yielding

$$\mathbf{W} = \mathbf{I}_N \otimes \mathbf{Q}_v^{-1}, \quad (4.6)$$

The solution $\hat{\mathbf{x}}$ provides an estimate of both the initial *EE* pose $\hat{\zeta}_0$ and the measurement poses $\hat{\zeta}_k$ for $k = 1, \dots, N$. The optimization problem is solved using an iterative nonlinear solver, which may rely on the Jacobian and either an approximation or the exact Hessian of the cost function $f(\mathbf{x})$. Denoting by $\mathbf{J} = \partial \mathbf{r} / \partial \mathbf{x} \in \mathbb{R}^{mN \times 6(N+1)}$ the residual Jacobian, the gradient of the cost function is given by

$$\frac{\partial f}{\partial \mathbf{x}} = \mathbf{J}^T \mathbf{W} \mathbf{r}. \quad (4.7)$$

The corresponding Hessian can be expressed as

$$\frac{\partial^2 f}{\partial \mathbf{x}^2} = \mathbf{J}^T \mathbf{W} \mathbf{J} + \sum_{k=1}^N \sum_{j=1}^m w_{j,k} r_{j,k} \frac{\partial^2 r_{j,k}}{\partial \mathbf{x}^2}, \quad (4.8)$$

where $w_{j,k}$ is the diagonal element of $\mathbf{W}$ relative to $r_{j,k}$ (i.e., the residual correspondent to the j -th sensor measurement in the k measurement pose, see Eqs. (4.2) and (4.1)). The first term corresponds to the Gauss–Newton approximation, while the second term is neglected in Gauss–Newton-type solvers.

Considering Eqs. (4.1) and (4.2), each residual $r_{j,k}$ depends at most on two pose variables, namely the initial pose ζ_0 and the corresponding measurement pose ζ_k . As

a consequence, the residual Jacobian $\mathbf{J}$ exhibits a sparse block structure, which can be assembled by stacking the pose derivatives of the modeled quantities $\mathbf{y}_k(\boldsymbol{\zeta}_k)$ as:

$$\mathbf{J} = \begin{bmatrix} -\mathbf{J}_{y_0} & \mathbf{J}_{y_1} & \mathbf{0}_{m \times 6} & \dots & \mathbf{0}_{m \times 6} \\ -\mathbf{J}_{y_0} & \mathbf{0}_{m \times 6} & \mathbf{J}_{y_2} & \dots & \mathbf{0}_{m \times 6} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{J}_{y_0} & \mathbf{0}_{m \times 6} & \mathbf{0}_{m \times 6} & \dots & \mathbf{J}_{y_N} \end{bmatrix} \quad (4.9)$$

where $\mathbf{J}_{y_k} = \partial \mathbf{y}_k / \partial \boldsymbol{\zeta}_k \in \mathbb{R}^{m \times 6}$. For residuals associated with absolute measurements, the dependence on the initial pose $\boldsymbol{\zeta}_0$ vanishes, resulting in zero rows in the corresponding $-\mathbf{J}_{y_0}$ block, while the derivatives with respect to $\boldsymbol{\zeta}_k$ remain nonzero.

The sparsity structure of the residual Jacobian directly reflects the functional dependence of each residual on the pose variables and carries over to the second-order derivatives. In particular, each residual $r_{j,k}$ depends only on the measurement pose $\boldsymbol{\zeta}_k$ for absolute measurements, and on both the initial pose $\boldsymbol{\zeta}_0$ and the corresponding measurement pose $\boldsymbol{\zeta}_k$ for relative measurements. As a consequence, the derivative $\partial^2 r_{j,k} / \partial \boldsymbol{\zeta}_a \partial \boldsymbol{\zeta}_b$ is nonzero only if $\boldsymbol{\zeta}_a$ and $\boldsymbol{\zeta}_b$ are either both equal to $\boldsymbol{\zeta}_k$, with $k = 0, \dots, N$. For this reason, the Hessian of each residual contains at most (when the corresponding measurement is relative) two nonzero 6×6 blocks. The nonzero second-order derivatives are directly given by the Hessians of the modeled quantities $y_{j,k}(\boldsymbol{\zeta}_k)$, as detailed below.

For relative measurements, the residual $r_{j,k}$ depends on both the initial pose $\boldsymbol{\zeta}_0$ and the corresponding measurement pose $\boldsymbol{\zeta}_k$. Accordingly, the only nonzero second-order derivatives are

$$\frac{\partial^2 r_{j,k}}{\partial \boldsymbol{\zeta}_0^2} = -\frac{\partial^2 y_{j,0}}{\partial \boldsymbol{\zeta}_0^2}, \quad \frac{\partial^2 r_{j,k}}{\partial \boldsymbol{\zeta}_k^2} = \frac{\partial^2 y_{j,k}}{\partial \boldsymbol{\zeta}_k^2}, \quad (\text{relative measurements}). \quad (4.10)$$

For absolute measurements, the residual $r_{j,k}$ depends only on the measurement pose $\boldsymbol{\zeta}_k$, and therefore the only nonzero second-order derivative is

$$\frac{\partial^2 r_{j,k}}{\partial \boldsymbol{\zeta}_k^2} = \frac{\partial^2 y_{j,k}}{\partial \boldsymbol{\zeta}_k^2}, \quad (\text{absolute measurements}). \quad (4.11)$$

The specific expressions depend on the kinematic relationships introduced in Sections 2.1.2 and 2.3.2.

The availability of the residual Hessians also enables the use of higher-order iterative solvers. In particular, Halley's method exploits a second-order Taylor approximation of the residual vector $\mathbf{r}$ to improve convergence when the initial guess is relatively inaccurate. Following [104], at iteration i a predictor increment is first computed via a Gauss-Newton step,

$$\delta \mathbf{x}_{\text{GN}}^{(i)} = \left(\mathbf{J}^{(i)T} \mathbf{W} \mathbf{J}^{(i)} \right)^{-1} \mathbf{J}^{(i)T} \mathbf{W} \mathbf{r}^{(i)}. \quad (4.12)$$

This predictor is then used to construct a corrected Jacobian $\mathbf{J}_H^{(i)}$ that accounts for second-order residual terms. Denoting by $\nabla^2 r_j^{(i)}$ the Hessian of the j -th residual component, the correction is assembled as

$$\mathbf{J}_H^{(i)} = \mathbf{J}^{(i)} - \frac{1}{2} \left[\nabla^2 r_1^{(i)} \delta \mathbf{x}_{\text{GN}}^{(i)} \quad \dots \quad \nabla^2 r_{mN}^{(i)} \delta \mathbf{x}_{\text{GN}}^{(i)} \right]^T, \quad (4.13)$$

where the residual Hessians can be directly obtained from Eqs. (4.10) and (4.11). The update $\delta \mathbf{x}^{(i)}$, such that $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \delta \mathbf{x}^{(i)}$ is then obtained by solving a damped least-squares problem based on $\mathbf{J}_H^{(i)}$:

$$\delta \mathbf{x}^{(i)} = \left(\mathbf{J}_H^{(i)T} \mathbf{W} \mathbf{J}_H^{(i)} + \lambda \mathbf{I} \right)^{-1} \mathbf{J}_H^{(i)T} \mathbf{W} \mathbf{r}^{(i)}, \quad (4.14)$$

The damping parameter $\lambda > 0$ is adjusted to enforce a monotonic reduction of the cost function f : if $f(\mathbf{x}^{(i+1)}) < f(\mathbf{x}^{(i)})$, λ is decreased by a factor of 10; otherwise, λ is increased by the same factor and the update is recomputed until a cost reduction is achieved.

4.2 Data acquisition for UACDPRs

The methods reported here aim at reducing data acquisition time, maintaining the same accuracy for the resulting initial pose, compared to previous studies [62]. A preliminary study for rapid data acquisition performed manually by an operator is reported in Sec. 4.2.1. While in Sec. 4.2.2, two data acquisition algorithms aimed at increasing the degree of automation while maintaining rapid and accurate self-calibration are proposed. In all cases, the initial-pose estimation problem is formulated as in Sec. 4.1. Experimental validation is conducted on a 4-cable UACDPR prototype, demonstrating the proposed methods' effectiveness in reducing calibration time while maintaining accuracy.

4.2.1 Manual method

In this technique, to collect the necessary data, the motion of the *EE* is induced manually, keeping cables taut through low-level tension control, while varying the *EE*'s position and orientation within the workspace. We assume that the UACDPR is equipped with incremental encoders on both the motors and swivel axes, allowing to measure swivel angle and cable length variations $\Delta \boldsymbol{\sigma}^*$ and $\Delta \mathbf{l}^*$, along with an *AHRS* sensor for the orientation measure $\boldsymbol{\epsilon}^*$ fixed to the platform.

Considering the formulation described in Sec. 4.1, the residual array $\mathbf{r}$ is built as:

$$\mathbf{r}_{l\sigma\epsilon} = [\mathbf{r}_l^T \mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(2n+3)}, \quad (4.15)$$

where:

$$\begin{aligned} \mathbf{r}_\sigma &= \begin{bmatrix} \boldsymbol{\sigma}(\zeta_1) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta \boldsymbol{\sigma}_1^* \\ \vdots \\ \boldsymbol{\sigma}(\zeta_N) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta \boldsymbol{\sigma}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, & \mathbf{r}_\epsilon &= \begin{bmatrix} \boldsymbol{\epsilon}(\zeta_1) - \boldsymbol{\epsilon}_1^* \\ \vdots \\ \boldsymbol{\epsilon}(\zeta_m) - \boldsymbol{\epsilon}_N^* \end{bmatrix} \in \mathbb{R}^{3N}, \\ \mathbf{r}_l &= \begin{bmatrix} \mathbf{l}(\zeta_1) - \mathbf{l}_0(\zeta_0) - \Delta \mathbf{l}_1^* \\ \vdots \\ \mathbf{l}(\zeta_N) - \mathbf{l}_0(\zeta_0) - \Delta \mathbf{l}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}. \end{aligned} \quad (4.16)$$

The arrays $\boldsymbol{\sigma}(\zeta_i)$ and $\mathbf{l}(\zeta_i)$ in $\mathbf{r}_\sigma$ and $\mathbf{r}_l$ are computed using inverse kinematics (Eqs. (2.44) and (2.13)).

4.2.1.1 Data acquisition procedure

The data acquisition algorithm aims at ensuring that each *DoF* of the *EE* is varied during the measurements, so the problem (4.4) is always well conditioned. The procedure can be summarized in two phases:

1. *Startup and tensioning*: the machine is switched on and the cables are controlled to reach a pre-determined tension set-point τ_0 . Consequently, the *EE* will move to a static configuration ζ_0 which will be known with several sources of uncertainties: due to unmodelled effects, at each machine startup, for the same tension control set-points, the attained end-effector pose will likely be different. The data collection of $\Delta \mathbf{l}_k^*$, $\Delta \boldsymbol{\sigma}_k^*$ and $\boldsymbol{\epsilon}_k^*$ starts once the *EE* reaches the equilibrium pose and all the residual oscillations resulting from the free motion are damped. The equilibrium pose of the platform can be stabilized with greater efficiency through the assistance of the operator. The tension setpoint will be kept for the entire duration of the self-calibration procedure;
2. *Manual excitation*: an operator manually varies the pose of the platform inside the workspace, such that all the configuration variables are properly excited. The values of $\Delta \mathbf{l}_k^*$, $\Delta \boldsymbol{\sigma}_k^*$ and $\boldsymbol{\epsilon}_k^*$ with $k = 1, \dots, N$ are recorded.

4.2.1.2 Tentative solution generation

The generation of a tentative solution $\mathbf{x}_{guess}$ for the problem 4.4 relies on the data acquisition process structure. First, a guess for the initial pose $\zeta_{0,guess}$ is obtained by solving the direct static problem described in Eq. (2.33) with the given tension τ_0 , resulting in an equilibrium pose likely close to the actual one, though possibly affected by unmodeled effects. From this equilibrium pose, cable lengths $\mathbf{l}_{0,guess}$ and swivel angles $\boldsymbol{\sigma}_{0,guess}$ are determined through inverse kinematics (cfr. Eq. 2.9 and 2.13). By adding the cable lengths and angle variations measured during the manual excitation phase, subsequent poses' cable lengths and swivel angles, denoted as $\mathbf{l}_i^* = \mathbf{l}_{0,guess} + \Delta \mathbf{l}_i^*$ and $\boldsymbol{\sigma}_i^* = \boldsymbol{\sigma}_{0,guess} + \Delta \boldsymbol{\sigma}_i^*$ are obtained. These variables, along with rotation data from the *AHRS*, are then used for pose estimation with the algorithm detailed in [50]. Due to the errors associated with $\mathbf{l}_{0,guess}$ and $\boldsymbol{\sigma}_{0,guess}$, the consequently determined poses are affected by errors as well.

4.2.1.3 Experiments and results

The efficacy of the data acquisition procedure was validated through experiments conducted on a 4-cable 6-*DoF* UACDPR, with the optimization algorithm implemented in MATLAB. The geometrical and inertial properties of the robot are presented in table 4.1, where $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are the unit vectors along the *Oxyz* axes. The platform mass is $m = 12.996$ kg and the vector that indicates the position of the center of mass with respect the mobile frame is $\mathbf{s}' = [0, 0, 0.173]^T$ m.

The solution $\hat{\mathbf{x}}$ of Eq.(4.4) is obtained using the MATLAB® routine *fminunc* using the BFGS-Newton algorithm. This routine incorporates termination criteria for the function value (set to 10^{-3}), the variables vector (set to 10^{-6}), and the infinity norm of the gradient (set to 10^{-4}). To speed up the solution, the analytical formulation of the gradient $\nabla f(\mathbf{x})$ is used as an input parameter for the solver resulting in an average

i	$r_i[\text{m}]$	$\mathbf{d}_i[\text{m}]$	${}^P\mathbf{a}'_i[\text{m}]$	x_i	y_i	z_i
1	0.03	$[1.068 \ -1.1155 \ 1.839]^T$	$[0.075 \ -0.075 \ 0.280]^T$	$-\mathbf{k}$	$\mathbf{j}$	$\mathbf{i}$
2	0.03	$[1.068 \ 1.1155 \ 1.839]^T$	$[0.075 \ 0.075 \ 0.280]^T$	$-\mathbf{k}$	$\mathbf{j}$	$\mathbf{i}$
3	0.03	$[-1.068 \ 1.1155 \ 1.839]^T$	$[-0.075 \ 0.075 \ 0.280]^T$	$-\mathbf{k}$	$-\mathbf{j}$	$-\mathbf{i}$
4	0.03	$[-1.068 \ -1.1155 \ 1.839]^T$	$[-0.075 \ -0.075 \ 0.280]^T$	$-\mathbf{k}$	$-\mathbf{j}$	$-\mathbf{i}$

Table 4.1: UACDPR properties

	$[\text{mm},^\circ]$	$\ \mathbf{e}_p\ [\text{m}]$	$\ \mathbf{e}_e\ [\circ]$	$t_c[\text{s}]$	$f(\mathbf{Z}_{sol})$
ζ_0	$[-341 \ 30 \ 423 \ -0.74 \ 7.58 \ 0.79]^T$	/	/	/	/
$\zeta_{0,sol,A}$	$[-333 \ 17 \ 410 \ -1.21 \ 7.53 \ -1.37]^T$	23.9	2.20	106	294.52
$\zeta_{0,sol,B}$	$[-333 \ 17 \ 409 \ -1.21 \ 7.53 \ -1.39]^T$	24.4	2.23	110	294.53
$\zeta_{0,sol,C}$	$[-334 \ 17 \ 406 \ -1.21 \ 7.53 \ -1.40]^T$	23.9	2.20	97	294.66
$\zeta_{0,sol,D}$	$[-333 \ 17 \ 409 \ -1.21 \ 7.53 \ -1.41]^T$	24.69	2.25	147	294.53
$\zeta_{0,sol,E}$	$[-334 \ 19 \ 409 \ -1.19 \ 7.51 \ -1.34]^T$	23.36	2.18	153	294.95
$\zeta_{0,sol,F}$	$[-282 \ 47 \ 410 \ -1.00 \ 7.51 \ 1.05]^T$	63.6	0.40	124	505.21

Table 4.2: Experimental Results

optimization time of about 112 seconds. The starting tension vector used for the experimentation is $\boldsymbol{\tau}_0 = [60 \ 60 \ 60 \ 60]^T \text{N}$, and the corresponding equilibrium pose computed through statics is $\zeta_{0,guess} = [0 \ 0 \ 617 \ 0 \ 0 \ 0]^T [\text{mm}, \circ]$. The sampling frequency of the measures was 2Hz, and a total of $m = 221$ samplings was recorded. To assess the accuracy of the initial pose estimation algorithm, the real initial pose ζ_0 was determined through a manual homing procedure of the *EE*. This homing procedure involves positioning the platform against a mechanical support where cable lengths are known and calibrated.

The value of the real initial pose is reported in the first line of table 4.2, while different results of the self-calibration algorithm $\zeta_{0,sol,A}, \dots, \zeta_{0,sol,F}$ obtained by using the same dataset but different initial guesses are shown below, to emphasize the importance of the latter. Solution *A* is obtained by computing $\zeta_{0,guess}$, called $\zeta_{0,guess,A}$ and the subsequent guess poses needed by $\mathbf{x}_{guess}$, as detailed in Sec.4.2.1.2, and is expected to converge to a global minimum with limited computational time. The guess for solution from *B* to *E* differs from *A* in how $\zeta_{0,guess}$ is computed. A position error norm ranging from 100 to 400 mm is randomly added to $\zeta_{0,guess}$, and the subsequent guess poses are computed as in Sec. 4.2.1.2. Lastly, the guess of solution *F* had the form $\mathbf{x}_{guess} = [\zeta_{0,guess,A}, \dots, \zeta_{0,guess,A}]$ to underline how a severely wrong guess would impact the convergence on a local minimum, instead of the global one. In the last third and fourth columns the position error norm and orientation error norm of solutions *A* to *F* with respect to the real pose ζ_0 are reported. In the fifth and sixth columns are reported the computational time of the self-calibration algorithm t_c and the final value of the cost function $f(\hat{\mathbf{x}})$ for each solution. The self-calibration algorithm yields positioning and orientation error results that are comparable to those of the state-of-the-art methodologies as referenced by Ida et al. in [62], yet it offers the advantages of requiring less time for data acquisition and processing. Solution *C* is obtained more rapidly as the random error provided to the initial guess has the beneficial effect of bringing the guess closer to the solution. Nonetheless, all solutions from *A* to *E* are equivalent to the numerical algorithm tolerances and are global

minima. On the contrary, solution F is a local minima.

4.2.2 Automatic methods

We consider *UACDPRs* equipped with multiple internal sensors for state estimation and motion control. In particular, an incremental encoder on the i -th swivel pulley measures variations in the swivel angles $\Delta\sigma_i^*$ relative to the angular position at startup $\sigma_{i,0}$. An *AHRS*, rigidly attached to the platform with its reference frame coincident with the *EE* mobile frame, provides roll and pitch angles (ϕ^* and θ^*) relative to a local-earth frame (east-north-up) coinciding with $Oxyz$, as well as the relative yaw variation $\Delta\chi^*$ from the startup configuration χ_0 . Additionally, an incremental encoder in the winch motor tracks the motor angle displacement $\Delta\theta_{m,i}^*$ relative to the startup value $\theta_{m,i,0}$. When the i -th winch is designed with a constant and known transmission ratio K between cable-length variation and motor displacement (as assumed here), the cable-length variation can be indirectly estimated as $\Delta l_i^* = K\Delta\theta_{m,i}^*$ [53]. Finally, a load cell is integrated into each transmission to measure the i -th cable tension τ_i^* .

The choice of this sensor set is based on three main reasons. First, previous studies [50, 105] demonstrated the effectiveness of incremental encoders and *AHRSs* for accurate pose estimation in *UACDPRs*. Therefore, it is reasonable to adopt this configuration as the available internal sensor suite, since these sensors will also be used for state estimation after calibration. Second, cable-tension measurements, whether obtained directly through load cells or estimated from motor currents [44], are typically employed for control and safety purposes, such as ensuring that all cables remain taut during operation. Since these measurements are already available in the control loop, they can be naturally included in the self-calibration formulation to enhance pose estimation accuracy. Third, the use of multiple sensors helps mitigate errors introduced by the simplified modeling assumptions described in Chapter 2, where cables are modeled as straight lines and the swivel pulleys are assumed to remain coplanar with the corresponding cable directions.

If other internal sensing systems are available, such as laser distance sensors for directly measuring cable elongation [106], they can be easily integrated into our approach, which is fully general. Indeed, including additional sensors may yield faster and more accurate results.

Two self-calibration algorithms are considered, each utilizing a different sensor set and indexed by $S \in \{\tau l \sigma \epsilon, l \sigma \epsilon\}$. The first algorithm ($S = \tau l \sigma \epsilon$) uses all the available sensors, including those measuring cable tensions, swivel-angle variations, cable-length variations, roll, pitch, and yaw variations. The second algorithm ($S = l \sigma \epsilon$) excludes cable tension measurements and only relies on the other sensors. The motivation for considering these two algorithms lies in the set of equations available for the self-calibration process, which directly influences the data acquisition strategy. On the one hand, maximizing the ratio between the number of equations and the number of unknowns is desirable, and this can be achieved by incorporating all available sensors, including load cells. However, using load cell measurements inherently requires relying on static equilibrium equations, as dynamic equations would introduce additional unknowns (namely, pose derivatives), counteracting the goal of maximizing the equation-to-unknowns ratio. As a consequence, the calibration process must rely on static *EE* poses, potentially leading to longer data acquisition times due to the need for

self-energy dissipation. Excluding load cell measurements simplifies the calibration process by relying only on geometric constraints. This allows for rapid movements and faster data acquisition at the cost of a lower equation-to-unknowns ratio, which may, in turn, reduce the accuracy of calibration results.

Using the formulation described in Sec. 4.1, the residual array $\mathbf{r}$ is thus composed of different combinations of residuals depending on the sensor set that is used:

$$\mathbf{r}_{\tau l \sigma \epsilon} = [\mathbf{r}_\tau^T \mathbf{r}_l^T \mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(3n+3)}, \quad \mathbf{r}_{l \sigma \epsilon} = [\mathbf{r}_l^T \mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(2n+3)}, \quad (4.17)$$

where:

$$\mathbf{r}_\sigma = \begin{bmatrix} \boldsymbol{\sigma}(\zeta_1) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta \boldsymbol{\sigma}_1^* \\ \vdots \\ \boldsymbol{\sigma}(\zeta_N) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta \boldsymbol{\sigma}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad \mathbf{r}_\epsilon = \begin{bmatrix} \phi(\zeta_1) - \phi_1^* \\ \theta(\zeta_1) - \theta_1^* \\ \chi(\zeta_1) - \chi_0(\zeta_0) - \Delta \chi_1^* \\ \vdots \\ \phi(\zeta_N) - \phi_N^* \\ \theta(\zeta_N) - \theta_N^* \\ \chi(\zeta_N) - \chi_0(\zeta_0) - \Delta \chi_N^* \end{bmatrix} \in \mathbb{R}^{3N}, \quad (4.18)$$

$$\mathbf{r}_l = \begin{bmatrix} \mathbf{l}(\zeta_1) - \mathbf{l}_0(\zeta_0) - \Delta \mathbf{l}_1^* \\ \vdots \\ \mathbf{l}(\zeta_N) - \mathbf{l}_0(\zeta_0) - \Delta \mathbf{l}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad \mathbf{r}_\tau = \begin{bmatrix} \boldsymbol{\tau}(\zeta_1) - \boldsymbol{\tau}_1^* \\ \vdots \\ \boldsymbol{\tau}(\zeta_N) - \boldsymbol{\tau}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad (4.19)$$

in which the superscript $(\cdot)^*$ indicates the measured quantities. The arrays $\boldsymbol{\sigma}(\zeta_i)$ and $\mathbf{l}(\zeta_i)$ in $\mathbf{r}_\sigma$ and $\mathbf{r}_l$ are computed using inverse kinematics (Eqs. (2.44) and (2.13)). Whereas, $\boldsymbol{\tau}(\zeta_i)$ in $\mathbf{r}_\tau$ can be computed from (2.33) through a generic left-inverse Ξ^{-T} of the structure matrix:

$$\boldsymbol{\tau}(\zeta_i) = \Xi^{-T}(\zeta_i) \mathbf{f}(\zeta_i). \quad (4.20)$$

To define a specific inverse of Ξ^T , we partition it into two blocks, namely $\Xi^T = [\Xi_c \ \Xi_f]^T$, where $\Xi_c \in \mathbb{R}^{n \times n}$ and $\Xi_f \in \mathbb{R}^{n \times (6-n)}$. The corresponding left-inverse is given by:

$$\Xi^{-T} = [\Xi_c^{-T} \ \mathbf{0}_{n \times (6-n)}], \quad (4.21)$$

where $\mathbf{0}_{n \times (6-n)}$ is a null matrix with n rows and $(6-n)$ columns.

The first objective of the automatic data-acquisition methods is to ensure well-distributed measurement poses within the static workspace of the robot. This helps condition the numerical solution to the nonlinear problem in Eq. (4.4), thus increasing the likelihood of convergence from an initial pose guess to the actual initial pose. The second objective is to define a control strategy and motion planning approach to move the *EE* to the measurement poses while satisfying the following requirements:

1. the motion control cannot rely on sensor reference values, as they are unknown at startup (their determination is the goal of the algorithm) or they are affected by high uncertainties;
2. in spite of requirement 1, the control strategy has to bring the *UACDPR* close enough to some ideal pre-determined measurement poses;
3. motion planning and control must keep tension within a lower and an upper bound, to prevent slackness and avoid cable breakage or exceeding actuator torque limits;

4. since the first calibration method relies on static equations, the *EE* must be brought to equilibrium configurations, reducing residual oscillations and enabling rapid data acquisition.

4.2.2.1 Selection of the measurement poses

Numerous theoretical methods have been proposed to determine optimal measurement configurations for calibration based on observability indices [107]. These approaches include discrete optimization over a predefined set of poses [108, 109], introducing deviations from ideal poses for improved robustness [110], and online iterative pose selection to maximize observability [59]. However, the direct application of these methods is challenging, since most of them assume that the robot can reach any *EE* pose, which is true for overconstrained *CDPRs* (e.g., [109, 110]) and, more generally, for non-underactuated architectures [107, 108]. In contrast, *UACDRs* can only partially control the *EE* pose due to inherent actuation limitations [111], so that observability criteria for fully actuated systems must be adapted, which is beyond the scope of this paper. For this reason, we adopt a more pragmatic, heuristic-based strategy.

For the method involving static equations, we select equispaced positions along the Cartesian axes within the workspace to ensure a uniform distribution of calibration configurations. This approach adapts the methodology originally proposed in [112] for overconstrained *CDPRs*, which ensures that measurement poses are well distributed in space and leads to a well-conditioned *NWLS* problem. To implement this approach, we first evaluate the robot's static workspace to determine its feasible boundaries, using the method described in Sec. 2.5 (Fig. 4.1a). Then, we select translational bounds $\mathbf{p}_{lb}$ and $\mathbf{p}_{ub}$ that define the corners of a parallelepiped inscribed in the static workspace. A regular grid of n_x , n_y , and n_z points is computed across the position intervals $[\mathbf{p}_{lb,x}, \mathbf{p}_{ub,x}]$, $[\mathbf{p}_{lb,y}, \mathbf{p}_{ub,y}]$, and $[\mathbf{p}_{lb,z}, \mathbf{p}_{ub,z}]$. Figure 4.1b shows an example with $n_x = n_y = n_z = 3$. At each position, we apply the method detailed in Sec. 2.5 to solve the modified *IGSP* and determine the orientation that satisfies static equilibrium. The resulting N poses are stored in the array $\mathbf{x}_{ideal}$, and their associated cable tensions, computed using Eq. (2.62), are stored as future control setpoints in the matrix $\mathbf{T}_{setpoint} = [\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_N]$. This procedure ensures that each measurement pose corresponds to a valid and controllable static equilibrium.

For the second self-calibration method, which does not rely on static equilibrium, we apply a similar principle in the time domain. Constant-velocity trajectories are executed across the workspace, and data are sampled at uniform time intervals. This approach aims to approximate a uniform spatial distribution of measurement poses. Although it does not explicitly optimize observability indices, prior studies suggest that calibration can still be effective if the poses are sufficiently numerous and well distributed in the workspace [4, 113]. Observability-based pose selection maximizes the sensitivity of the residuals to the unknown parameters, which improves convergence when only a limited number of poses is used. However, comparable results may be achieved using a sufficient number of uniformly or randomly sampled configurations, particularly near workspace boundaries [108]. Therefore, the proposed approach offers a pragmatic solution that can ensure acceptable accuracy, provided that the control strategy maintains the kinematic constraints (i.e., cable tensions remain within the prescribed bounds).

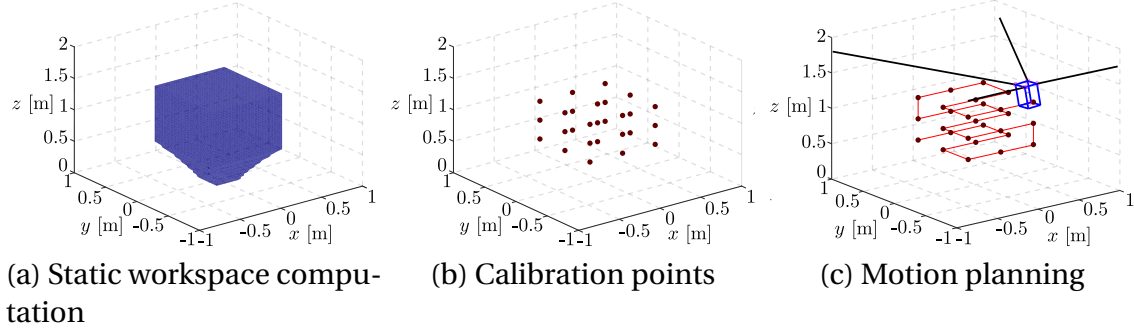


Figure 4.1: Calibration poses selection and control planning

4.2.2.2 Data-acquisition procedure

Recalling requirements 1 through 4, it is not feasible to move the *EE* between equilibrium measurement poses and simultaneously suppress residual oscillations using conventional trajectory shaping techniques for vibration damping, such as input-shaped trajectories (e.g., [111]). These methods require accurate joint-position feedback, which in turn depends on a calibrated reference for cable lengths. Therefore, they are only applicable after the initial self-calibration process has been completed.

Furthermore, robust control strategies for UACDPRs that can cope with large geometric uncertainties (such as those present prior to calibration) remain an open research challenge. For this reason, a conservative control approach was adopted in this study, prioritizing safety and constraint preservation performance.

Once the static equilibrium measurement poses are defined and the UACDPR is powered on, joint-space force control is used to drive the *EE*. Assuming that the *EE* starts within the static workspace, the initial cable tensions are recorded, and a smooth transition is commanded toward a desired tension vector $\boldsymbol{\tau}_0$. In this context, we define $\hat{\boldsymbol{\tau}}(t)$ as the time-varying tension reference used to guide the *EE* between each equilibrium configuration $\boldsymbol{\tau}_i \in \mathbf{T}_{setpoint}$ ². While any static configuration corresponding to $\boldsymbol{\tau}_0$ is theoretically valid, a convenient “home” pose may be chosen, and $\boldsymbol{\tau}_0$ accordingly computed via static analysis. This initial motion is essential for generating the initial guess used in Section 4.2.2.3.

After allowing sufficient time for the *EE* to reach equilibrium, data logging begins. The tension setpoints in $\mathbf{T}_{setpoint}$ are then sequentially applied, with a waiting period introduced after each setpoint $\boldsymbol{\tau}_i$ is reached. Static equilibrium at each pose is detected by monitoring the ripple in the measured cable tensions. When the amplitude of the ripple falls below the load cell’s uncertainty threshold, the system is considered to have reached a steady state, and residual oscillations are assumed to be negligible.

The sequence of equilibrium poses is chosen to minimize transition time (Fig. 4.1c). Although there is no explicit control of the *EE* pose between tension setpoints, pure force control maintains active cable constraints, ensuring requirement 3 is met. Since this method is sensitive to load-cell noise, accurate calibration of load cells is crucial. If the measured cable tensions deviate from their true value, the actual equilibrium configurations of the *EE* may differ from the expected ones, leading to incor-

²In this work, a 5th-degree polynomial is used to generate each cable-tension setpoint. This approach does not guarantee that the *EE* remains strictly inside the static workspace during transitions, but the motion profiles can be experimentally adjusted to keep constraints active (e.g., by increasing the transition duration).

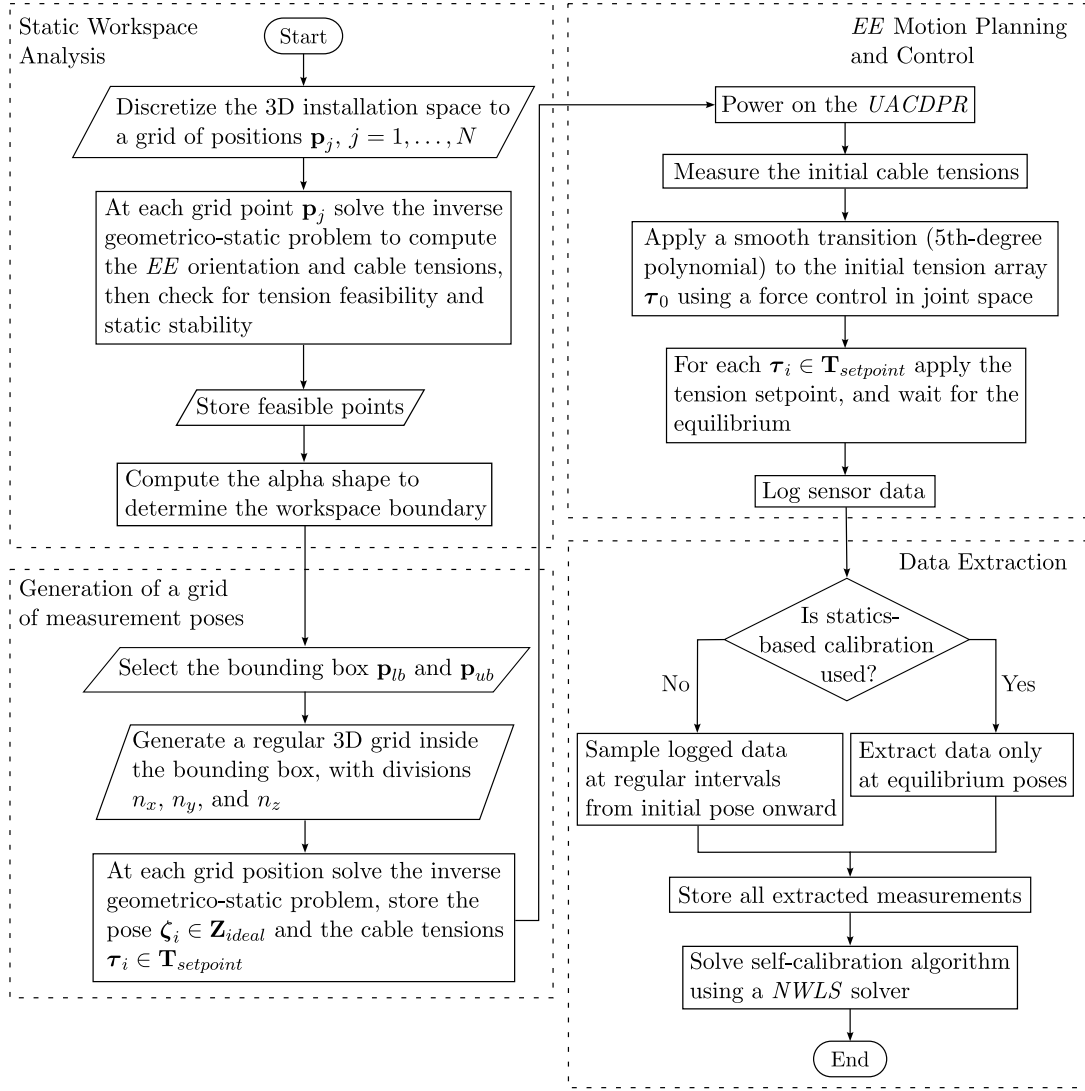


Figure 4.2: Flowchart of the data-acquisition algorithm

rect assumptions about static constraints. This, in turn, may affect the accuracy of the self-calibration algorithm, as the constraints used to estimate the initial pose may no longer be valid. Moreover, unaccounted tension errors can lead to situations where the EE moves outside the static workspace, further compromising the reliability of the collected measurements.

To minimize total data acquisition time, we adopted a trial-and-error approach to tune the trajectory profiles, striking a balance between fast transitions (enabled by higher velocities) and short oscillation damping periods. This tuning process is architecture-specific and needs to be performed only once for a given UACDPR. If the system's geometric and inertial properties remain unchanged, the identified motion parameters can be reused across all subsequent startups, ensuring that the proposed self-calibration procedure remains fully automatic in routine operation. This assumption aligns with the current objective of automating initial-pose self-calibration, which is based on a fixed mechanical architecture. However, it would need to be reconsidered if the method were extended to the self-calibration of geometric parameters, as might be required in reconfigurable UACDPRs.

The data acquisition method for the two automatic self-calibration algorithms dif-

fers in the data extraction procedure. For the static-equation-based method, only equilibrium pose measurements from $\mathbf{x}_{ideal}$ are used, identified by the stationary values of measured tensions in the pre-defined setpoints $\mathbf{T}_{setpoint}$. In contrast, for the method that does not rely on static equations, any measure following the one corresponding to the initial pose is a valid candidate for the calibration measurement set. Therefore, any sample time that is a multiple of the logging interval can be selected to collect a sufficient number of measurements N and to uniformly distribute measurement poses in the workspace.

The complete acquisition workflow for both self-calibration strategies is summarized in the flowchart in Figure 4.2

4.2.2.3 Initial guess

The self-calibration problem (4.4) is solved using a gradient-based optimizer, whose convergence depends on the quality of the initial guess $\mathbf{x}_{guess} = [\zeta_{0,guess}^T, \zeta_{1,guess}^T, \dots, \zeta_{N,guess}^T]^T$. For both self-calibration algorithms, the initial guess for the pose $\zeta_{0,guess}$ is obtained by solving the direct static problem (2.33) using the given tension setpoint τ_0 . For the measurement poses $\zeta_{1,guess}^T, \dots, \zeta_{N,guess}^T$, a suitable tentative solution for algorithm with $j = \tau l \sigma \epsilon$ is given by the ideal equilibrium poses included in the array $\mathbf{x}_{ideal}$. For the self-calibration with $j = l \sigma \epsilon$, instead, a suitable initial tentative solution for $\zeta_{1,guess}^T, \dots, \zeta_{N,guess}^T$ can be found through the following passages:

1. from the initial equilibrium pose $\zeta_{0,guess}$, cable lengths $\mathbf{l}_{0,guess}$ and swivel angles $\sigma_{0,guess}$ are obtained through inverse kinematics;
2. estimates of cable lengths $\mathbf{l}_i^*$ and swivel angles σ_i^* relative to the i -th measurement pose are obtained by adding the i -th cable-length variations and swivel-angle variations measured during the EE motion to $\mathbf{l}_{0,guess}$ and $\sigma_{0,guess}$, namely $\mathbf{l}_i^* = \mathbf{l}_{0,guess} + \Delta \mathbf{l}_i^*$ and $\sigma_i^* = \sigma_{0,guess} + \Delta \sigma_i^*$; this computation is made for all measurement poses, thus obtaining $\mathbf{l}_1^*, \dots, \mathbf{l}_k^*$ and $\sigma_1^*, \dots, \sigma_k^*$;
3. the measurement pose vector $\mathbf{Z}_{guess}$ is obtained performing a pose estimation for each measurement pose using the corresponding reference values of $\mathbf{l}_i^*$, σ_i^* , and ϵ_i^* , through the sensor-fusion algorithm described in [50].

4.2.2.4 Experiments and results

The effectiveness of the automatic data acquisition methods was assessed through experiments on a spatial 4-cable UACDPR prototype at the IRMA L@B laboratory of the University of Bologna, shown in Fig. 4.3. The self-calibration algorithms were tested offline in a MATLAB environment. Table 4.3 reports the constant geometric parameters of the UACDPR, where $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the unit vectors of the local pulley frames aligned with the $Oxyz$ axes. The coordinates of vectors $\mathbf{d}_i$ and $\mathbf{a}_i'$ are expressed in $Oxyz$ and $Pxyz$ respectively. These values were measured using a meter tool with an uncertainty of 1 mm. The EE mass, measured with a weighing scale, is $m = 12.992$ kg, and the vector that points from P to G , obtained by the CAD model of the platform and expressed in the mobile frame, is $\mathbf{s}' = [0 \ 0 \ 0.173]^T$ m. The sensors employed are shown in Fig. 4.3b, where the platform (A in Fig. 4.3a) and the winch (B in Fig. 4.3a) are shown in detail. Specifically, cable lengths were estimated by using 20-bit incremental encoders

i	r_i [m]	$\mathbf{d}_i$ [m]	$\mathbf{a}'_i$ [m]	x_i	y_i	z_i
1	0.03	$[1.065 \ -1.155 \ 1.839]^T$	$[0.075 \ -0.075 \ 0.266]^T$	$-\mathbf{k}$	$\mathbf{j}$	$\mathbf{i}$
2	0.03	$[1.066 \ 1.155 \ 1.839]^T$	$[0.075 \ 0.075 \ 0.266]^T$	$-\mathbf{k}$	$\mathbf{j}$	$\mathbf{i}$
3	0.03	$[-1.066 \ 1.155 \ 1.839]^T$	$[-0.075 \ 0.075 \ 0.266]^T$	$-\mathbf{k}$	$-\mathbf{j}$	$-\mathbf{i}$
4	0.03	$[-1.065 \ -1.155 \ 1.839]^T$	$[-0.075 \ -0.075 \ 0.266]^T$	$-\mathbf{k}$	$-\mathbf{j}$	$-\mathbf{i}$

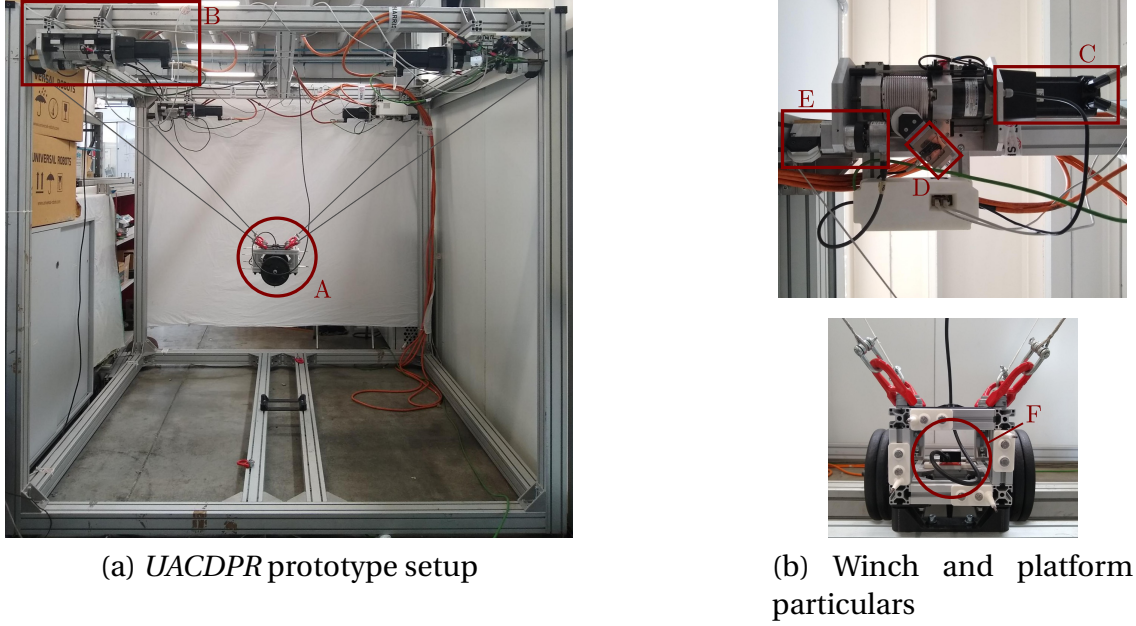
Table 4.3: Geometrical parameters of the 4-cable *UACDPR* prototype

Figure 4.3: Experimental setup

with an uncertainty of $\pm 0.144^\circ$ on each motor axis (C in Fig. 4.3b), whereas swivel angles were measured by 16-bit incremental encoders with an uncertainty of $\pm 0.03^\circ$ (E in Fig. 4.3b), mounted directly on the swivel axes of pulleys. Cable tensions were measured using load cells inside the transmission (D in Fig. 4.3b) with a full scale (*FS.*) of 100 kg, and accuracy of 0.02% *FS.* *EE* Euler angles (*ZYX* convention) were measured using an *Xsens MTi-630 AHRS* fixed to the platform (F in Fig. 4.3b). The uncertainty is $\pm 0.2^\circ$ for roll and pitch and $\pm 1^\circ$ for yaw. The *AHRS* was configured to set the startup value of the yaw measure to 0, as the north reference value given by the magnetometer is not reliable due to the high electro-magnetic disturbances in the *UACDPR* environment. For each sensor set S , the measurement-noise covariance is taken as diagonal, $Q_v = \text{diag}(\sigma_1^2, \dots, \sigma_m^2)$, with entries selected from Table 4.4 according to the included measurements (cable-length variations, swivel-angle variations, *AHRS* components, and cable tensions). To measure the real *EE* initial pose $\zeta_{0,real} = [\mathbf{p}_{0,real}^T \ \boldsymbol{\epsilon}_{0,real}^T]^T$ and

σ_l^2 [m ²]	σ_σ^2 [deg ²]	σ_ϵ^2 [deg ²]	σ_τ^2 [N ²]
0.01	1	2	10

Table 4.4: Sensor noise model parameters.

use it as ground truth for the self-calibration results, a manual homing procedure was performed before each test. The platform was positioned against a mechanical sup-

i	$\zeta_{ideal,i}$ [m, rad]	$\tau_{setpoint,i}$ [N]
1	$[0.40 \ 0.40 \ 1.20 \ 0.08 \ -0.09 \ 0.01]^T$	$[118.48 \ 170.83 \ 115.57 \ 82.98]^T$
2	$[-0.40 \ 0.40 \ 1.20 \ 0.08 \ 0.09 \ -0.00]^T$	$[82.98 \ 115.57 \ 170.83 \ 118.48]^T$
3	$[-0.40 \ -0.40 \ 1.20 \ -0.08 \ 0.09 \ 0.00]^T$	$[115.49 \ 82.97 \ 118.54 \ 170.73]^T$
4	$[0.40 \ -0.40 \ 1.20 \ -0.08 \ -0.09 \ -0.00]^T$	$[170.73 \ 118.54 \ 82.97 \ 115.49]^T$
5	$[0.40 \ -0.40 \ 0.60 \ -0.15 \ -0.15 \ -0.01]^T$	$[81.18 \ 55.90 \ 31.01 \ 53.86]^T$
6	$[-0.40 \ -0.40 \ 0.60 \ -0.15 \ 0.15 \ 0.01]^T$	$[53.86 \ 70.04 \ 81.18 \ 31.01]^T$
7	$[-0.40 \ 0.40 \ 0.60 \ 0.15 \ 0.15 \ -0.01]^T$	$[31.01 \ 53.87 \ 81.21 \ 55.87]^T$
8	$[0.40 \ 0.40 \ 0.60 \ 0.15 \ -0.15 \ 0.01]^T$	$[55.87 \ 81.21 \ 53.87 \ 31.01]^T$

Table 4.5: Ideal measurement poses and tension setpoints, with $n_x = n_y = n_z = 2$

port where the nominal values of cable lengths, swivel angles, and yaw were previously measured with a high-precision photogrammetry system.

After the static workspace analysis, the translational bounds were set as:

$$\mathbf{p}_{lb} = [-0.4 \ -0.4 \ 0.6]^T \text{ m} \quad \text{and} \quad \mathbf{p}_{ub} = [0.4 \ 0.4 \ 1.2]^T \text{ m}. \quad (4.22)$$

The choice of the number of measurement poses k was guided by the main objective of this work: achieving automatic self-calibration in reduced time, without compromising accuracy. For the same reason, our evaluation is mainly based on the calibration time. To analyze the effect of different measurement-pose sets on the initial-pose self-calibration, two spatial grids were discretized using different values of n_x , n_y , and n_z . In the first grid, $n_x = n_y = n_z = 2$, resulting in a total of $k = 8$ equilibrium poses. In the second grid, $n_x = n_y = n_z = 3$, yielding $k = 27$ poses. The equilibrium poses $\mathbf{Z}_{ideal}$ and tension setpoints $\mathbf{T}_{setpoint}$ for $k = 2$ are reported in Table 4.5 as an example, while results for $k = 27$ are omitted as they do not provide additional insights.

The initial tension setpoint was set close to the first component τ_1 of $\mathbf{T}_{setpoint}$ (which is the same for the two grids), at $\tau_0 = [100.00 \ 100.00 \ 100.00 \ 100.00]$ N. The data logging rate for all sensors was 100 ms, balancing file size and data resolution. The self-calibration problem (4.4) was solved using a PC with a 13th generation Intel i7 processor and 32 GB of RAM, distributing the computation across 14 cores. For both $j = \tau l \sigma \epsilon$ and $j = l \sigma \epsilon$ the *fmincon* MATLAB routine with a trust-region-reflective algorithm was used with a termination tolerance of 10^{-10} on the function value, of 10^{-8} on the first-order optimality, and of 10^{-10} on the optimization variables. The gradient was analytically formulated to improve computational efficiency, as in [60], and the initial guess was generated using the methods described in Sec. 4.2.2.3. A graphical representation of the measured tensions, along with the reference trajectory $\hat{\mathbf{r}}$ used to transition between the equilibrium configurations, is shown in Fig. 4.4.

The results of the two automatic self-calibration methods are reported in Table 4.6, where the following data are specified:

- the used sensor set j , and thus, the related self-calibration algorithm;
- the total self-calibration time t_{sc} , composed by the data acquisition time and initial-pose estimation time;
- the number of pose measurements k used in the algorithm;

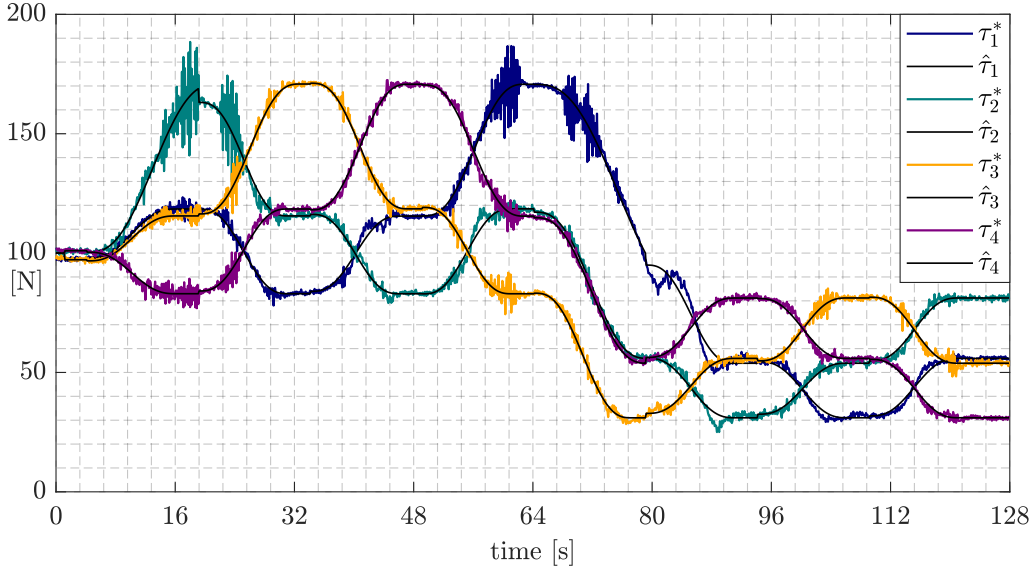


Figure 4.4: Sampled and reference tensions with $k = 8$. The black curves, with superscript ($\hat{\cdot}$), represent the tension reference applied during transitions, while the colored curves are the measured tensions, with superscript ($\cdot^*$).

Trial	j	t_{sc} [min]	k	ϵ_p [mm]	ϵ_r [deg]
<i>a</i>	$\tau l \sigma \epsilon$	2.23	8	12	1.24
<i>b</i>	$l \sigma \epsilon$	2.24	19	23	5.44
<i>c</i>	$\tau l \sigma \epsilon$	10.04	27	7.2	3.14
<i>d</i>	$l \sigma \epsilon$	10.02	128	22	2.86

Table 4.6: Initial-pose self-calibration algorithm results

- the resulting initial position error norm $\epsilon_p = \|\mathbf{p}_{0,real} - \mathbf{p}_{0,sol}\|$;
- the resulting initial orientation error $\epsilon_r = |\alpha_{0,real} - \alpha_{0,sol}|$, where α_0 is the rotation angle in the axis-angle parameterization of the initial orientation. The angle α is obtained from the rotation matrix $\mathbf{R}(\epsilon)$ as:

$$\alpha = \arccos\left(\frac{\mathbf{R}_{1,1}(\epsilon) + \mathbf{R}_{2,2}(\epsilon) + \mathbf{R}_{3,3}(\epsilon) - 1}{2}\right). \quad (4.23)$$

Trials *a* and *b* share the same dataset, corresponding to the data acquisition in which the *EE* transitioned between 8 equilibrium poses. Similarly, trials *c* and *d* share a dataset where the *EE* followed the path with 27 equilibrium poses. Therefore, the data acquisition time for trials *a* and *b* is the same and equal to $t_{da} = 2.13$ min, whereas $t_{da} = 5.11$ min for trials *c* and *d*. The total self-calibration time t_{sc} reported in Table 4.6 consists of both the data acquisition time t_{da} and the computational time required by the solver t_{opt} , which depends on the number of poses k used in the optimization. For trials *b* and *d*, where cable tension measurements were not used, the pose sampling rate from the logged dataset was chosen such that the number of extracted poses k yielded an optimization time t_{opt} approximately equal to that of their tension-based counterparts (trials *a* and *c*). This ensured a fair comparison between methods that do and do not utilize load-cell measurements.

Incorporating tension measurements ($j = \tau l \sigma \epsilon$) substantially improves position accuracy. For instance, using only 8 equilibrium poses, the error drops from 23 mm

(trial *b*) to 12 mm (trial *a*). Similarly, for $k = 27$, it decreases from 22 mm (trial *d*) to 7.2 mm (trial *c*). This trend confirms that static equilibrium constraints and cable tension measurements significantly enhance positional accuracy. The effect on orientation accuracy is more difficult to interpret. When comparing trials *a* and *b*, the inclusion of tension measurements reduces the orientation error from 5.44° to 1.24° , showing a clear benefit. However, this trend does not persist when increasing the number of poses. In fact, trial *c* (tension-based) results in a worse orientation estimate (3.14°) than trial *a* (1.24°). This may be due to less favorable pose distribution or greater noise influence in the larger dataset. Conversely, in the kinematic-only approach, increasing k from 19 (trial *b*) to 128 (trial *d*) halves the orientation error (5.44° to 2.86°), as expected. These results suggest that the impact of the number of measurement poses on orientation accuracy is method-dependent: while the kinematic-only method benefits from more poses, the tension-based method does not exhibit a clear trend and may be sensitive to dataset-specific factors.

Overall, the tension-based method (*a*, *c*) achieves better position accuracy while requiring significantly fewer measurements than the kinematic-only method (*b*, *d*). This is particularly noteworthy because the tension-based method relies on a simplified physical model (neglecting cable sag and elasticity) and uses cable-tension measurements, which typically carry greater uncertainty than kinematic signals. Yet, despite these limitations, the inclusion of tension data and equilibrium constraints consistently improves the positional estimate. It is also important to note that, although internal statics has been previously used in self-calibration problems [62], to our knowledge, direct use of cable tension measurements for initial-pose self-calibration in *UACDPRs* is novel. This work therefore provides the first quantitative comparison of statics- vs. kinematics-based methods using real tension data. To obtain self-calibration results comparable to those reported in [60], which achieved a best-case accuracy of $\varepsilon_p = 23.36$ mm and $\varepsilon_r = 2.18^\circ$ using a fast but manual procedure, the kinematic-only approach requires at least $k = 128$ poses and a total calibration time of over 10 minutes. In contrast, the tension-based method achieves better performance with just $k = 8$ and with the same self-calibration duration. These findings emphasize the importance of exploiting static configurations and cable-tension measurements to enable fast and accurate automatic self-calibration in *UACDPRs*.

4.3 Algorithm and parameter analysis

This section presents the analysis methodology adopted to systematically evaluate the influence of key design parameters on the performance of initial-pose self-calibration in *CDPRs*. In detail, in section 4.3.1 a preliminary analysis on deployable *OCDPRs* is included to analyze the specific impact of the sensor set and number of measurement configuration chosen for the self-calibration. The aim is to evaluate how the choice of sensors impacts the accuracy of the initial-pose estimation, particularly when cable length variations are either unavailable or highly uncertain, as when nonlinear and uncertain transmissions are used (which is often the case in deployable *CDPRs*). The three considered sensor sets are:

1. incremental encoders on swivel pulleys and an Attitude and Heading Reference System (AHRS) for *EE* orientation;

2. the same sensors of item 1, augmented with incremental encoders on winch motors to estimate cable lengths;
3. the same sensors of item 1, enhanced with load cells to measure cable tensions.

Whereas the analysis presented in Sec. 4.3.3 aims at comparing different choices of the number of measurement poses N , the sensor set S , and the numerical solver K , in terms of their impact on calibration accuracy and robustness. Using the general formulation of the self-calibration problem introduced in Sec. 4.1, the methodology for choosing the set of measurement poses is defined in Sec. 4.3.2. In Sec. 4.3.3, a comprehensive Monte Carlo simulation framework is defined, in which multiple realizations of the calibration process are performed under prescribed levels of parameter uncertainty and measurement noise. For each combination of (N, S, K) , a set of performance indices is computed and statistically characterized through their mean values and variances over the Monte Carlo trials, which constitute the basis for the comparison presented in the following section.

4.3.1 Preliminary analysis on parameter selection for deployable *OCDPRs*

In this analysis, we assume that the *CDPR* can be equipped with multiple sensors for state estimation and motion control [60]. Specifically, an incremental encoder on the i -th swivel pulley measures variations in the swivel angles $\Delta\sigma_i$, relative to the angular position at startup. An *AHRS*, rigidly attached to the platform with its reference frame coincident with the *EE* mobile frame, provides roll and pitch angles (ϕ and θ) relative to a local-earth frame (east-north-up) coinciding with $Oxyz$, as well as the relative yaw variation ($\Delta\chi$) from the startup configuration. Load cells are incorporated into the transmissions to measure cable tensions, while incremental encoders in the winch motors track motor angle displacements $\Delta\theta_{m_i}$'s relative to the startup value. Weight minimization is a key design factor in deployable *CDPRs*, since both winches and cable transmissions are integrated into the moving platform to maximize portability. In turn, this requires to use simpler mechanical transmissions, which cannot ensure a constant or predictable linear relationship between $\Delta\theta_{m_i}$ and the cable length variation Δl_i [53].

We consider here three measurement sets for initial-pose self-calibration, indexed by $S = [\sigma\epsilon, l\sigma\epsilon, \tau\sigma\epsilon]$ and corresponding to: (i) swivel angles and *AHRS* measurements, (ii) cable lengths, swivel angles, and *AHRS* measurements, and (iii) cable tensions, swivel angles, and *AHRS* measurements. Since swivel angles can be reliably and economically measured in *CDPRs*, and prior work [50] has demonstrated that the *AHRS* significantly improves pose estimation, they are always included in all measurement sets. The main research question is whether these sensors are enough or cable lengths and tensions, which are affected by potentially high errors, can still be beneficial due to sensor redundancy.

Using the formulation described in Sec. 4.1, the residual array $\mathbf{r}$ is composed of different combinations of residuals depending on the sensor set that is used:

$$\mathbf{r}_{\sigma\epsilon} = [\mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(n+3)}, \quad \mathbf{r}_{l\sigma\epsilon} = [\mathbf{r}_l^T \mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(2n+3)}, \quad (4.24)$$

$$\mathbf{r}_{\tau\sigma\epsilon} = [\mathbf{r}_\tau^T \mathbf{r}_\sigma^T \mathbf{r}_\epsilon^T]^T \in \mathbb{R}^{N(2n+3)}, \quad (4.25)$$

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{a}_8$
$\begin{bmatrix} 2.44 \\ 0.31 \\ -1.75 \end{bmatrix}$	$\begin{bmatrix} 2.41 \\ -0.378 \\ -1.8 \end{bmatrix}$	$\begin{bmatrix} -2.41 \\ -0.378 \\ -1.8 \end{bmatrix}$	$\begin{bmatrix} -2.44 \\ 0.31 \\ -1.75 \end{bmatrix}$	$\begin{bmatrix} 2.41 \\ 0.378 \\ 1.8 \end{bmatrix}$	$\begin{bmatrix} 2.44 \\ -0.31 \\ 1.75 \end{bmatrix}$	$\begin{bmatrix} -2.44 \\ -0.31 \\ 1.75 \end{bmatrix}$	$\begin{bmatrix} -2.41 \\ 0.378 \\ 1.8 \end{bmatrix}$
$\mathbf{d}'_1$	$\mathbf{d}'_2$	$\mathbf{d}'_3$	$\mathbf{d}'_4$	$\mathbf{d}'_5$	$\mathbf{d}'_6$	$\mathbf{d}'_7$	$\mathbf{d}'_8$
$\begin{bmatrix} 0.302 \\ -0.115 \\ 0.203 \end{bmatrix}$	$\begin{bmatrix} 0.263 \\ 0.193 \\ 0.114 \end{bmatrix}$	$\begin{bmatrix} -0.263 \\ 0.193 \\ 0.114 \end{bmatrix}$	$\begin{bmatrix} -0.302 \\ -0.115 \\ 0.203 \end{bmatrix}$	$\begin{bmatrix} 0.246 \\ -0.116 \\ -0.096 \end{bmatrix}$	$\begin{bmatrix} 0.245 \\ 0.174 \\ -0.139 \end{bmatrix}$	$\begin{bmatrix} -0.245 \\ 0.174 \\ -0.139 \end{bmatrix}$	$\begin{bmatrix} -0.246 \\ -0.116 \\ -0.096 \end{bmatrix}$

Table 4.7: Geometrical parameters [m] of the *IRMA-8 OCDPR*

where:

$$\mathbf{r}_\sigma = \begin{bmatrix} \boldsymbol{\sigma}(\zeta_1) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta\boldsymbol{\sigma}_1^* \\ \vdots \\ \boldsymbol{\sigma}(\zeta_N) - \boldsymbol{\sigma}_0(\zeta_0) - \Delta\boldsymbol{\sigma}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad \mathbf{r}_\epsilon = \begin{bmatrix} \phi(\zeta_1) - \phi_1^* \\ \theta(\zeta_1) - \theta_1^* \\ \chi(\zeta_1) - \chi_0(\zeta_0) - \Delta\chi_1^* \\ \vdots \\ \phi(\zeta_N) - \phi_N^* \\ \theta(\zeta_N) - \theta_N^* \\ \chi(\zeta_N) - \chi_0(\zeta_0) - \Delta\chi_N^* \end{bmatrix} \in \mathbb{R}^{3N}, \quad (4.26)$$

$$\mathbf{r}_l = \begin{bmatrix} \mathbf{l}(\zeta_1) - \mathbf{l}_0(\zeta_0) - \Delta\mathbf{l}_1^* \\ \vdots \\ \mathbf{l}(\zeta_N) - \mathbf{l}_0(\zeta_0) - \Delta\mathbf{l}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad \mathbf{r}_\tau = \begin{bmatrix} \boldsymbol{\tau}(\zeta_1) - \boldsymbol{\tau}_1^* \\ \vdots \\ \boldsymbol{\tau}(\zeta_N) - \boldsymbol{\tau}_N^* \end{bmatrix} \in \mathbb{R}^{Nn}, \quad (4.27)$$

in which the superscript $(\cdot)^*$ indicates the measured quantities. The arrays $\boldsymbol{\sigma}(\zeta_i)$ and $\mathbf{l}(\zeta_i)$ in $\mathbf{r}_\sigma$ and $\mathbf{r}_l$ are computed using inverse kinematics (Eqs. (2.44) and (2.13)). Whereas, $\boldsymbol{\tau}(\zeta_i)$ in $\mathbf{r}_\tau$ is computed considering a closed-form tension distribution [35] assuming the system to be force controlled.

4.3.1.1 Simulation Results

To evaluate the performance of the initial-pose self-calibration method described in the previous section, we conducted simulations on the *IRMA-8* deployable *OCDPR* prototype synthesized in [29] (Fig. 4.5). The constant geometric parameters³ are listed in Table 4.7, with an *EE* mass of 40 kg and a center of mass located at $\mathbf{s}' = [0; 0; 0]^T$ m in the mobile frame. Since the prototype operates with a constant orientation due to its workspace being optimized for this working mode [29], a constant identity rotation matrix is assumed throughout the calibration process. A preliminary constant orientation *WFW* analysis is performed to determine the upper and lower *EE* position bounds ($\mathbf{p}_{lb}$ and $\mathbf{p}_{ub}$), thus ensuring that calibration poses remain within a feasible workspace:

$$\mathbf{p}_{lb} = [-1.2 \ -0.2 \ -1.6]^T, \quad \mathbf{p}_{ub} = [1.2 \ 0.2 \ 0.8]^T. \quad (4.28)$$

Within these limits, different sets of initial pose and measurement poses, $\mathbf{x}_{ideal} \in \mathbb{R}^{6(N+1)}$, were selected for $N = 8, 18, 27, 48, 64$, to assess the impact of the configuration number on calibration accuracy, with poses evenly distributed along the principal axes. Ideal sensor measurements are obtained via inverse kinematics and statics, with

³These parameters may differ from the same presented in other sections of this thesis due to relocation or redesign of the cable attachment points.

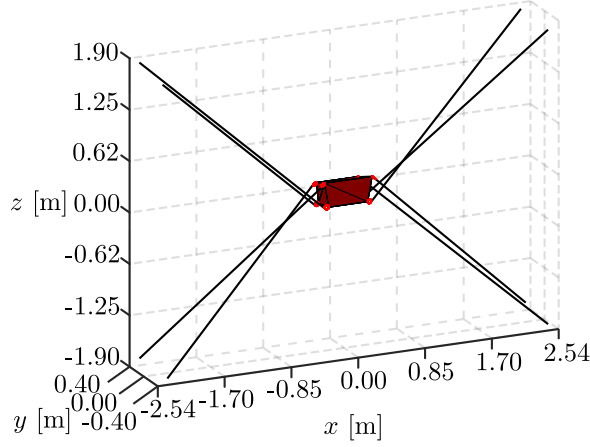


Figure 4.5: Schematic of the prototype

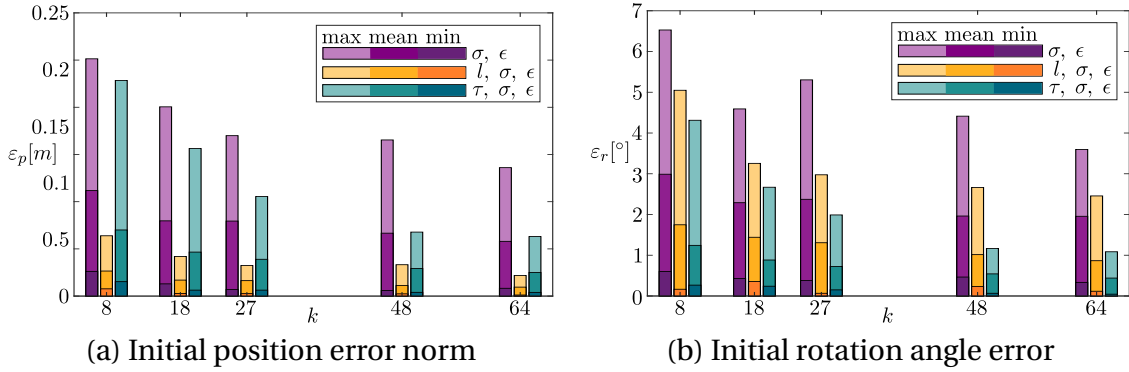


Figure 4.6: Calibrated initial pose errors

artificial random noise simulating sensor uncertainties and model simplifications. For statistical robustness, 50 trials were conducted for each S and N , and the results were stored. The self-calibration solution, $\mathbf{x}_{sol,S,N}$, was then obtained for each S and N by using an *Interior-Point* optimization solver applied to the weighted least squares formulation in Eq. (4.4). Assuming that the CDPR control can bring the *EE* close enough to the ideal poses $\mathbf{x}_{ideal}$, they are used as initial guesses for the solver.

The initial-pose self-calibration results are illustrated in Figure 4.6 for different sensor sets. The considered error metrics are the initial position error norm, $\varepsilon_p = \|\mathbf{p}_{0,ideal} - \mathbf{p}_{0,sol}\|$, and the initial orientation error, $\varepsilon_r = |\alpha_{0,initial} - \alpha_{0,sol}|$, where α_0 is the rotation angle in the axis-angle parameterization of the initial orientation. The bar charts include superimposed minimum, mean, and maximum error values (progressively with darker colors), illustrating the spread of calibration accuracy. The minimum values of ε_p and ε_r are consistently close to zero, indicating that at least some trials achieved highly accurate calibration. A clear decreasing trend in position errors is observed in Fig. 4.6a as the number of calibration poses increases, while Fig. 4.6b confirms that the mean orientation error also reduces with a larger number of poses. Comparing the different sensor sets, it is evident that using only swivel angles and end-effector orientation results in the highest calibration errors. The inclusion of cable tensions or cable lengths significantly improves accuracy, even in the presence of sensor noise. In particular, cable-length measurements exceed all other approaches in estimating the initial position, while cable tensions offer better accuracy for the initial orientation. Moreover, compared to others, the load cell sensor set seems to be less

sensitive to noise when k increases, as the maximum position and orientation error are significantly reduced. These findings confirm that an accurate initial pose calibration requires cable-length measurements and that increasing the number of calibration poses enhances accuracy by mitigating measurement uncertainties.

4.3.2 Measurement Pose Selection

For a given number of measurement poses N , several methods can be used to select informative configurations within the *CDPR* workspace [108, 110, 114, 115]. In this work, the number of measurement poses N is treated as the primary parameter of influence, while the pose-selection strategy itself is assumed to be fixed and validated. Under this assumption, the impact of the specific selection method is considered secondary with respect to the effect of varying N . Accordingly, the adopted pose-selection method is regarded as a representative and repeatable baseline, rather than as an optimal solution to the measurement-configuration problem.

For each value of N and sensor set S , a set of ideal measurement poses $\mathbf{x}_{ideal}$ is determined offline using the constrained optimization method proposed in [114], selected for its effectiveness and ease of implementation. The method relies on a genetic algorithm (*GA*) to maximize an observability index O , defined as the inverse of the condition number of the weighted residual Jacobian:

$$\mathbf{x}_{ideal} = \arg\max_{\mathbf{x}} O(\mathbf{x}), \quad \text{s.t. } c(\mathbf{x}), \quad \text{with} \quad O(\mathbf{x}) = \frac{1}{\text{Cond}\left(\sqrt{\mathbf{W}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right)}. \quad (4.29)$$

The condition number is adopted as a scalar observability metric, as it captures the sensitivity of the estimation problem to measurement perturbations and is a well-established numerical index for calibration-oriented pose selection [116]. The constraint set $c(\mathbf{x})$ to be applied to Eq. (4.29) must ensure that all the measurement poses are within the *CDPR* workspace, which depends on the *CDPR* architecture. For this reason a workspace analysis is performed as a preliminary step to define admissible bounds on the *EE* pose components.

For *UACDPRs*, the workspace is computed following the approach described in Sec 2.5, where the modified inverse geometrico-static problem (*IGSP*) is solved to determine the set of reachable *EE* poses that satisfy both wrench-feasibility and stability conditions. In particular, static equilibrium poses are evaluated over a grid of *EE* positions, and configurations that do not satisfy the wrench-feasibility and stability conditions are discarded. Subsequently, lower and upper bounds on the *EE* position, $\mathbf{p}_{lb}$ and $\mathbf{p}_{ub}$ respectively, are defined to delimit a parallelepiped fully contained within the workspace. Since only a subset of the *EE* pose can be directly controlled in *UACDPRs*, the orientation bounds are not explicitly defined, but rather implicitly found through the modified inverse geometrico-static problem. Therefore, the constraint set $c(\mathbf{x})$ in Eq. (4.29) for *UACDPRs* is defined as

$$c(\mathbf{x}) = \begin{cases} \mathbf{p}_{lb} \leq \mathbf{p}_{k,ideal} \leq \mathbf{p}_{ub} \\ \boldsymbol{\epsilon}_{k,ideal} = \text{IGSP}(\mathbf{p}_{k,ideal}) \end{cases} \quad \forall k = 0, \dots, N. \quad (4.30)$$

For the implementation detail of $\text{IGSP}(\mathbf{p}_{k,ideal})$ please refer to [49].

For *OCDPRs*, a constant-orientation workspace with $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{const}$ is determined as in [36] by discretizing the installation volume and retaining only the positions that satisfy wrench-feasibility conditions. Subsequently, lower and upper bounds on the *EE*

position, $\mathbf{p}_{lb}$ and $\mathbf{p}_{ub}$, respectively, are defined to identify a parallelepiped fully contained within the constant-orientation workspace. Accordingly, the constraint set $c(\mathbf{x})$ in Eq. (4.29) for *OCDPRs* is defined as

$$c(\mathbf{x}) = \begin{cases} \mathbf{p}_{lb} \leq \mathbf{p}_{k,ideal} \leq \mathbf{p}_{ub} \\ \boldsymbol{\epsilon}_{k,ideal} = \boldsymbol{\epsilon}_{const} \end{cases} \quad \forall k = 0, \dots, N, \quad (4.31)$$

This choice reflects common operating conditions of *OCDPRs*, where many tasks (including calibration procedures) are executed at constant end-effector orientation.

4.3.3 Analysis of Parameter Influence

The performance of the initial-pose self-calibration procedure depends on several setup parameters, as discussed in Sec. 4.1. In this work, we focus on three categories of parameters that reflect the most critical design choices in practical implementations. Each parameter is discretized over a finite set of values, allowing for a systematic comparative analysis.

The first parameter is the number of measurement poses N , which determines the amount of data available to estimate the unknown initial pose. The value of N is varied as $N \in \{10, 20, 30, 40, 50\}$, covering a range that spans from minimally overdetermined systems, which allow for a fast data acquisition procedure but low amount of information, to highly overdetermined ones which lead to more accurate results⁴ at the cost of a longer procedure. The second parameter is the sensor set used to acquire measurements at each pose. Four sensor configurations are considered, each corresponding to a different subset of the residual components introduced in Sec. 4.1. The *LENGTH* configuration relies exclusively on cable-length variations $\Delta \mathbf{l}_k^*$ measured by relative encoders mounted on the motor winches. The *LENGTH-SWIVEL* configuration augments cable-length measurements with swivel-angle variations $\Delta \boldsymbol{\sigma}_k^*$, obtained from rotary encoders installed on the pulleys. The *LENGTH-AHRS* configuration combines cable-length variations $\Delta \mathbf{l}_k^*$ and yaw variations $\Delta \chi_k^*$ with absolute roll ϕ_k^* and pitch θ_k^* measurements provided by an *AHRS* mounted on the *EE*. Finally, the *LENGTH-SWIVEL-AHRS* configuration includes all the aforementioned measurements, namely $\Delta \mathbf{l}_k^*$, $\Delta \boldsymbol{\sigma}_k^*$, ϕ_k^* , θ_k^* , and $\Delta \chi_k^*$. The third parameter is the nonlinear solver employed to solve the optimization problem in Eq. (4.4). Three solvers are considered. The first is the Levenberg–Marquardt method [118, 119], which relies on a Gauss–Newton approximation of the Hessian. The second is a trust-region Newton method [120], which uses the exact Hessian of the cost function. The third is Halley’s method [104], which exploits second-order information to improve convergence when the tentative solution is far from the real one, which is often the case in self-calibration.

The self-calibration procedure is modeled as a stochastic process in which uncertainty arises from two independent sources. The first is the measurement noise $\mathbf{v}_y$, modeled as zero-mean Gaussian and defined as in Sec. 4.1. The second is the pose-reaching error $\mathbf{v}_\zeta$ to model inaccuracies in achieving the desired measurement poses during the data acquisition phase. Specifically, the actual measurement poses $\boldsymbol{\zeta}_{k,real}$

⁴It is worth noting that extensive datasets lead to nonlinear least-squares problems with large residual vectors, which may suffer from numerical ill-conditioning and amplification of round-off errors due to finite machine precision [117], potentially degrading estimation accuracy. Such cases are not considered in this work, as the number of measurement poses remains moderate.

($k = 1, \dots, N$) are generated by perturbing the ideal poses $\zeta_{k,ideal}$ as:

$$\zeta_{k,real} = \zeta_{k,ideal} + \mathbf{v}_\zeta. \quad (4.32)$$

Here, $\mathbf{v}_\zeta = [\mathbf{v}_p^T \ \mathbf{v}_e^T]^T$ represents a control error, modeled as independent, zero-mean, and uniformly distributed component-wise:

$$\mathbf{v}_p \sim U(-\delta_p, \delta_p), \quad \mathbf{v}_e \sim U(-\delta_e, \delta_e). \quad (4.33)$$

The parameters $\delta_p \in \mathbb{R}^3$ and $\delta_e \in \mathbb{R}^3$ denote maximum admissible position and orientation errors, respectively. A uniform distribution is adopted to model pose-reaching errors, as it is realistically achievable by a controller in practice and provides a conservative representation of control uncertainty without favoring specific error realizations.

To assess the influence of the aforementioned parameters on self-calibration performance, a Monte Carlo (MC) analysis is conducted. For each combination of parameters, M independent trials are performed. In each trial, random realizations of the measurement noise and pose-reaching error are generated, and the optimization problem in Eq. (4.4) is solved to obtain an estimate $\hat{\mathbf{x}}$ of the unknown variables.

Several performance indices are used to evaluate the outcome of the self-calibration. Recalling that $\hat{\mathbf{x}} = [\hat{\mathbf{p}}_0^T, \hat{\mathbf{e}}_0^T, \hat{\zeta}_1^T, \dots, \hat{\zeta}_N^T]^T$ is the solution of the self-calibration problem, estimation accuracy is quantified through the initial position error norm $\varepsilon_p = \|\hat{\mathbf{p}}_0 - \mathbf{p}_0\|$ and the initial orientation error⁵ $\varepsilon_r = |\hat{\alpha}_0 - \alpha_0|$, where α_0 denotes the rotation angle in the axis-angle representation of the initial orientation. The estimated rotation angle α can be computed from the rotation matrix $\mathbf{R}$ as:

$$\alpha = \arccos\left(\frac{\mathbf{R}_{1,1} + \mathbf{R}_{2,2} + \mathbf{R}_{3,3} - 1}{2}\right). \quad (4.34)$$

Computational efficiency is evaluated by measuring the computation time t_{comp} and the number of iterations n_{iter} required by the nonlinear solver to converge. Numerical consistency is assessed through the value of the cost function at the solution, normalized by the number of measurement poses as $f_{sol} = f(\hat{\mathbf{x}})/N$, allowing for fair comparisons across different values of N . Finally, to quantify the consistency of the simulation results, we define the inlier ratio IR as the percentage of trials remaining after the exclusion of outliers via the Interquartile Range (IQR) method [121]:

$$\text{IR} = \frac{M_{clean}}{M} \cdot 100, \quad (4.35)$$

where M_{clean} represents the number of trials classified as inliers. In detail, the IQR method was applied to the normalized final cost f_{sol} to identify and discard outliers. Ordering the data from the smallest to the largest, and dividing the data in four quartiles, where the first quartile (Q_1) is defined as the 25th percentile, where the lowest 25% data lies below this point. The second quartile (Q_2) is the median of a data set, thus 50% of the data lies below this point. The third quartile (Q_3) is the 75th percentile,

⁵Regarding the orientation error metric, the norm of the Euler-angle difference is not adopted, as it lacks a clear physical meaning due to the periodicity and possible discontinuities of Euler-angle representations [61]. Instead, the axis-angle representation provides a geometrically meaningful scalar measure of orientation error, corresponding to the magnitude of the relative rotation between two orientations.

where the lowest 75% data lies below this point. According to the standard IQR criterion, a trial is considered an inlier if its cost lies within the interval

$$[Q_1(f_{sol}) - 1.5IQR, Q_3(f_{sol}) + 1.5IQR], \quad (4.36)$$

where $Q_1(\cdot)$ and $Q_3(\cdot)$ denote the first and third quartiles, respectively, and $IQR = Q_3(f_{sol}) - Q_1(f_{sol})$. With this method, trials with very low f_{sol} , which likely identify better results, are discarded too. Thus making the analysis not unbalanced towards an unlikely better result. The number of inliers M_{clean} is then obtained as the cardinality of the corresponding set of trials.

The analysis procedure begins by loading the CDPR model and performing a workspace analysis to determine the constraint set for measurement-pose selection, as described in Sec. 4.3.2. The following steps are summarized in Alg. 1. For each combination of the number of measurement poses N , sensor set S , and nonlinear solver K , M independent MC trials are performed. Since the optimal measurement poses depend on both N and S , they are first computed using the GA-based method described in Sec. 4.3.2 (summarized in Alg. 1 with the function *SelectMeasurementPosesGA*). In each trial, the actual measurement poses and sensor outputs are generated by simulating pose-reaching errors and measurement noise. This step is summarized in Alg. 1 with the function *ComputeControlAndMeasureError* (see Alg. 2), where the actual measurement poses $\zeta_{k,real}$ are obtained by perturbing the ideal poses $\zeta_{k,ideal}$ with uniformly distributed pose-reaching errors through Eq. (4.32). The real reference quantities in the initial pose $\mathbf{l}_0$, σ_0 , and in the measurement poses $\mathbf{l}_k$, and σ_k (for all $k = 1, \dots, N$) are then generated by evaluating the CDPR inverse kinematics (summarized in the algorithm with the function *ComputeInverseKinematics*). Specifically, using Eqs. (2.9) and (2.13) for all $i = 1, \dots, n$, applied respectively to the initial pose $\zeta_{0,real}$ and to each measurement pose $\zeta_{k,real}$ ($k = 1, \dots, N$). Subsequently, the simulated sensor measurements are obtained for each k by computing the relative variations (only for relative measurements) and adding Gaussian noise, as detailed in Sec. 4.1. Finally, all the real poses and measurements are collected in the vectors:

$$\Delta \mathbf{l}^* = \begin{bmatrix} \Delta \mathbf{l}_1^* \\ \Delta \mathbf{l}_2^* \\ \vdots \\ \Delta \mathbf{l}_N^* \end{bmatrix}, \quad \Delta \sigma^* = \begin{bmatrix} \Delta \sigma_1^* \\ \Delta \sigma_2^* \\ \vdots \\ \Delta \sigma_N^* \end{bmatrix}, \quad \phi^* = \begin{bmatrix} \phi_1^* \\ \phi_2^* \\ \vdots \\ \phi_N^* \end{bmatrix}, \quad \theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \\ \vdots \\ \theta_N^* \end{bmatrix}, \quad \Delta \chi^* = \begin{bmatrix} \Delta \chi_1^* \\ \Delta \chi_2^* \\ \vdots \\ \Delta \chi_N^* \end{bmatrix}. \quad (4.37)$$

The self-calibration cost function, its gradient, and its Hessian are then constructed based on the current combination of the self-calibration parameters, and the generated measurements, following the steps summarized in function *GenerateSelfCalibrationFunction*, which implements the passages described in Sec. 4.1.

As an example, if the sensor set S is *LENGTH-SWIVEL-AHRS*, considering Eqs. (4.1)–(4.2), the residuals associated with the k -th measurement pose are defined as:

$$\mathbf{r}_{l,k} = \mathbf{l}(\zeta_k) - \mathbf{l}(\zeta_0) - \Delta \mathbf{l}_k^*, \quad \mathbf{r}_{\sigma,k} = \sigma(\zeta_k) - \sigma(\zeta_0) - \Delta \sigma_k^*, \quad \mathbf{r}_{e,k} = \begin{bmatrix} \phi(\zeta_k) - \phi_k^* \\ \theta(\zeta_k) - \theta_k^* \\ \chi(\zeta_k) - \chi(\zeta_0) - \Delta \chi_k^* \end{bmatrix}, \quad (4.38)$$

where $\mathbf{r}_{l,k}$, $\mathbf{r}_{\sigma,k}$, and $\mathbf{r}_{e,k}$ denote the residuals associated with cable-length variations, swivel-angle variations, and orientation measurements at the k -th pose, respectively.

Algorithm 1: Monte Carlo analysis for initial-pose self-calibration parameter influence

Data: $\mathbf{p}_{lb}, \mathbf{p}_{ub}, \boldsymbol{\epsilon}_{const}$ (only for OCDPRs), $\mathcal{N}$ (set of N), $\mathcal{S}$ (sensor-set list); $\mathcal{K}$ (solver list), M , $\boldsymbol{\delta}_p, \boldsymbol{\delta}_\epsilon, \mathbf{Q}_v$;

CDPR geometric and inertial parameters;

Result: Dataset of performance indices for each (N, S, K) and MC statistics

```

1  foreach  $N \in \mathcal{N}$  do
2    foreach  $S \in \mathcal{S}$  do
3       $\mathbf{x}_{ideal} \leftarrow \text{SelectMeasurementPosesGA}(N, S, \mathbf{p}_{lb}, \mathbf{p}_{ub}, \boldsymbol{\epsilon}_{const}, \mathbf{Q}_v)$  /* Generate ideal
        measurement poses */
4      foreach  $K \in \mathcal{K}$  do
5        for  $m = 1$  to  $M$  do
6          // Generate real poses and measurements
           $[\mathbf{x}_{real}, \Delta \mathbf{l}^*, \Delta \boldsymbol{\sigma}^*, \boldsymbol{\phi}^*, \boldsymbol{\theta}^*, \Delta \boldsymbol{\chi}^*] \leftarrow \text{ComputeControlAndMeasureError}(\mathbf{x}_{ideal}, S,$ 
             $N, \mathbf{Q}_v, \boldsymbol{\delta}_p, \boldsymbol{\delta}_\epsilon)$ 
          // Build residuals, cost, gradient, and Hessian according to
            Sec. 4.1
7           $[f(\mathbf{x}), \frac{\partial f}{\partial \mathbf{x}}, \frac{\partial^2 f}{\partial \mathbf{x}^2}] \leftarrow \text{GenerateSelfCalibrationFunction}(N, S, \Delta \mathbf{l}^*, \Delta \boldsymbol{\sigma}^*, \boldsymbol{\phi}^*, \boldsymbol{\theta}^*,$ 
             $\Delta \boldsymbol{\chi}^*)$ 
          // Solve the nonlinear optimization problem
8           $\mathbf{x}_0 \leftarrow \mathbf{x}_{ideal}$  /* Set initial guess */
9          if  $K = LM$  then
10              $(\mathbf{x}_{sol}, f_{sol}, n_{iter}, \text{exitflag}, t_{comp}) \leftarrow$ 
              LevenbergMarquardtSolver( $f(\mathbf{x}), \frac{\partial f}{\partial \mathbf{x}}, \mathbf{x}_0$ )
11          else if  $K = TRH$  then
12              $(\mathbf{x}_{sol}, f_{sol}, n_{iter}, \text{exitflag}, t_{comp}) \leftarrow \text{TrustRegionSolver}(f(\mathbf{x}), \frac{\partial f}{\partial \mathbf{x}}, \frac{\partial^2 f}{\partial \mathbf{x}^2}, \mathbf{x}_0)$ 
13          else
14              $(\mathbf{x}_{sol}, f_{sol}, n_{iter}, \text{exitflag}, t_{comp}) \leftarrow \text{HalleySolver}(f(\mathbf{x}), \frac{\partial f}{\partial \mathbf{x}}, \frac{\partial^2 f}{\partial \mathbf{x}^2}, \mathbf{x}_0)$ 
15          end
          // Compute performance indices for the trial
16           $\epsilon_p = \|\hat{\mathbf{p}}_0 - \mathbf{p}_{0,real}\|$  /* Position error */
17           $\epsilon_r = |\hat{\alpha}_0 - \alpha_{0,real}|$  /* Orientation error */
18          store( $N, S, K, \epsilon_p, \epsilon_r, t_{comp}, n_{iter}, f_{sol}, \text{exitflag}$ )
19        end
20      ComputeAndStoreStatisticsMC( $N, S, K$ ) /* Aggregate and save results for
        this  $(N, S, K)$  */
21    end
22  end
23 end

```

Algorithm 2: ComputeControlAndMeasureError

Input: $\mathbf{x}_{ideal}, S, N, \mathbf{Q}_v, \delta_p, \delta_\epsilon$
Output: $[\mathbf{x}_{real}, \Delta \mathbf{l}^*, \Delta \sigma^*, \phi^*, \theta^*, \Delta \chi^*]$

// Simulate pose-reaching errors

- 1 **for** $k = 0$ **to** N **do**
- 2 $\mathbf{v}_p \sim U(-\delta_p, \delta_p)$
- 3 $\mathbf{v}_\epsilon \sim U(-\delta_\epsilon, \delta_\epsilon)$
- 4 $\mathbf{v}_\zeta = [\mathbf{v}_p^T \mathbf{v}_\epsilon^T]^T$
- 5 $\zeta_{k,real} = \zeta_{k,ideal} + \mathbf{v}_\zeta$
- 6 **end**

// Simulate sensor measurements at the disturbed poses

- 7 $[\mathbf{l}_0, \sigma_0] \leftarrow \text{ComputeInverseKinematics}(\zeta_{0,real})$ /* Compute initial cable lengths and swivel angles */
- 8 **for** $k = 1$ **to** N **do**
- 9 $\mathbf{v}_l \sim \mathcal{N}(\mathbf{0}, \sigma_l^2 \mathbf{I}_{n \times n})$
- 10 $\mathbf{v}_\sigma \sim \mathcal{N}(\mathbf{0}, \sigma_\sigma^2 \mathbf{I}_{n \times n})$
- 11 $v_\phi \sim \mathcal{N}(0, \sigma_\phi^2)$
- 12 $v_\theta \sim \mathcal{N}(0, \sigma_\theta^2)$
- 13 $v_\chi \sim \mathcal{N}(0, \sigma_\chi^2)$
- 14 $[\mathbf{l}_k, \sigma_k] \leftarrow \text{ComputeInverseKinematics}(\zeta_{k,real})$ /* Compute modelled cable lengths and swivel angles */
- 15 $\Delta \mathbf{l}_k^* = \mathbf{l}_k - \mathbf{l}_0 + \mathbf{v}_l$
- 16 $\Delta \sigma_k^* = \sigma_k - \sigma_0 + \mathbf{v}_\sigma$
- 17 $\phi_k^* = \phi_{k,real} + v_\phi$
- 18 $\theta_k^* = \theta_{k,real} + v_\theta$
- 19 $\Delta \chi_k^* = \chi_{k,real} - \chi_{0,real} + v_\chi$
- 20 **end**
- 21 **return** $[\mathbf{x}_{real}, \Delta \mathbf{l}^*, \Delta \sigma^*, \phi^*, \theta^*, \Delta \chi^*]$

Collecting the residuals for all measurement poses $k = 1, \dots, N$ and grouping them by sensor type, the overall residual vector components are written as:

$$\mathbf{r}_l = \begin{bmatrix} \mathbf{l}(\zeta_1) - \mathbf{l}(\zeta_0) - \Delta \mathbf{l}_1^* \\ \mathbf{l}(\zeta_2) - \mathbf{l}(\zeta_0) - \Delta \mathbf{l}_2^* \\ \vdots \\ \mathbf{l}(\zeta_N) - \mathbf{l}(\zeta_0) - \Delta \mathbf{l}_N^* \end{bmatrix}, \quad \mathbf{r}_\sigma = \begin{bmatrix} \sigma(\zeta_1) - \sigma(\zeta_0) - \Delta \sigma_1^* \\ \sigma(\zeta_2) - \sigma(\zeta_0) - \Delta \sigma_2^* \\ \vdots \\ \sigma(\zeta_N) - \sigma(\zeta_0) - \Delta \sigma_N^* \end{bmatrix}, \quad \mathbf{r}_\epsilon = \begin{bmatrix} \phi(\zeta_1) - \phi_1^* \\ \theta(\zeta_1) - \theta_1^* \\ \chi(\zeta_1) - \chi(\zeta_0) - \Delta \chi_1^* \\ \phi(\zeta_2) - \phi_2^* \\ \theta(\zeta_2) - \theta_2^* \\ \chi(\zeta_2) - \chi(\zeta_0) - \Delta \chi_2^* \\ \vdots \\ \phi(\zeta_N) - \phi_N^* \\ \theta(\zeta_N) - \theta_N^* \\ \chi(\zeta_N) - \chi(\zeta_0) - \Delta \chi_N^* \end{bmatrix}. \quad (4.39)$$

Accordingly, the derivative of the residual vector w.r.t. the unknowns $\mathbf{x}$ is computed

analytically using the Jacobians derived in Eqs. (2.21) (2.22) (or (2.52)) as:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \begin{bmatrix} -\mathbf{J}_l(\zeta_0) & \mathbf{J}_l(\zeta_1) & \dots & \mathbf{0}_{n \times 6} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{J}_l(\zeta_0) & \mathbf{0}_{n \times 6} & \dots & \mathbf{J}_l(\zeta_k) \\ -\mathbf{J}_\sigma(\zeta_0) & \mathbf{J}_\sigma(\zeta_1) & \dots & \mathbf{0}_{n \times 6} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{J}_\sigma(\zeta_0) & \mathbf{0}_{n \times 6} & \dots & \mathbf{J}_\sigma(\zeta_k) \\ -\mathbf{J}_\chi(\zeta_0) & \mathbf{J}_\epsilon(\zeta_1) & \dots & \mathbf{0}_{3 \times 6} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{J}_\chi(\zeta_0) & \mathbf{0}_{3 \times 6} & \dots & \mathbf{J}_\epsilon(\zeta_k) \end{bmatrix} \in \mathbb{R}^{N(2n+3) \times 6(N+1)} \quad (4.40)$$

where $\mathbf{J}_\epsilon(\zeta_k) = \partial \mathbf{r}_{\epsilon,k} / \partial \zeta_k = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3}]$ and $\mathbf{J}_\chi(\zeta_0) = \partial \mathbf{r}_{\chi,k} / \partial \zeta_0$ is for all k :

$$\mathbf{J}_\chi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.41)$$

while the nonzero second-order derivatives of the residual components are computed using Eqs. (4.10) and (4.11). For instance, for residuals arising from the i -th cable length and swivel angle, the second derivatives with respect to the initial pose ζ_0 are computed through Eqs. (2.27) and (2.30):

$$\frac{\partial^2 r_{l_{k,i}}}{\partial \zeta_0^2} = -\frac{\partial^2 l_{i,0}}{\partial \zeta_0^2}, \quad \frac{\partial^2 r_{\sigma_{k,i}}}{\partial \zeta_0^2} = -\frac{\partial^2 \sigma_{i,0}}{\partial \zeta_0^2}, \quad \forall \quad k = 1, \dots, N, \quad (4.42)$$

in the same way, the second derivatives with respect to the k -th pose ζ_k are:

$$\frac{\partial^2 r_{l_{k,i}}}{\partial \zeta_k^2} = \frac{\partial^2 l_i}{\partial \zeta_k^2}, \quad \frac{\partial^2 r_{\sigma_{k,i}}}{\partial \zeta_k^2} = \frac{\partial^2 \sigma_i}{\partial \zeta_k^2}. \quad (4.43)$$

All the other second order derivatives of the components of $\mathbf{r}$ are null.

Then, the optimization problem is solved using the selected nonlinear solver. Finally, performance indices are computed for each trial, and statistics are aggregated and stored.

4.3.4 Results

The parameter and algorithm analysis introduced in Sec. 4.3.3 is applied to two different *CDPR* architectures, selected to represent distinct geometries and workspaces: (i) a suspended 8-cable *OCDPR* developed at the ARDC Lab⁶, and the IRMA-4U 4-cable *UACDPR* prototype developed at IrmaL@b⁷. For both systems, the analysis is first conducted in simulation via the Monte Carlo framework of Sec. 4.3.3, and a subset of the analyzed parameter and algorithm combinations is then evaluated experimentally on the corresponding prototypes.

The *OCDPR* moving platform has mass $m = 3.7$ kg and center of mass coincident with the end-effector frame, $\mathbf{s}' = [0 \ 0 \ 0]^T$ m. The geometric parameters $\mathbf{d}_i$ and $\mathbf{a}'_i$ are

⁶Aerospace, Robotics, Dynamics, and Control Lab, University of Minnesota, Minneapolis, MN, USA.

⁷Industrial Robotics, Mechatronics & Automation Laboratory @ Bologna, Bologna, Italy.

i	1	2	3	4	5	6	7	8
$\mathbf{d}_i$ [m]	$\begin{bmatrix} 1.667 \\ 0.849 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} 1.578 \\ 0.937 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} -1.459 \\ 1.395 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} -1.548 \\ 1.306 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} -0.626 \\ -0.860 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} -0.537 \\ -0.948 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} 1.539 \\ -0.992 \\ 2.423 \end{bmatrix}$	$\begin{bmatrix} 1.628 \\ -0.903 \\ 2.423 \end{bmatrix}$
${}^P\mathbf{a}'_i$ [m]	$\begin{bmatrix} 0.165 \\ -0.095 \\ 0.161 \end{bmatrix}$	$\begin{bmatrix} -0.095 \\ 0.165 \\ -0.194 \end{bmatrix}$	$\begin{bmatrix} 0.095 \\ 0.165 \\ 0.161 \end{bmatrix}$	$\begin{bmatrix} -0.165 \\ -0.095 \\ -0.194 \end{bmatrix}$	$\begin{bmatrix} -0.165 \\ 0.095 \\ 0.161 \end{bmatrix}$	$\begin{bmatrix} 0.095 \\ -0.165 \\ -0.194 \end{bmatrix}$	$\begin{bmatrix} -0.095 \\ -0.165 \\ 0.161 \end{bmatrix}$	$\begin{bmatrix} 0.165 \\ 0.095 \\ -0.194 \end{bmatrix}$

Table 4.8: Geometric parameters of the ARDC Lab *OCDPR*

i	1	2	3	4
$\mathbf{d}_i$ [m]	$\begin{bmatrix} 1.080 \\ -1.153 \\ 1.846 \end{bmatrix}$	$\begin{bmatrix} 1.080 \\ 1.153 \\ 1.846 \end{bmatrix}$	$\begin{bmatrix} -1.079 \\ 1.153 \\ 1.846 \end{bmatrix}$	$\begin{bmatrix} -1.078 \\ -1.153 \\ 1.846 \end{bmatrix}$
${}^P\mathbf{a}'_i$ [m]	$\begin{bmatrix} 0.075 \\ -0.075 \\ 0.266 \end{bmatrix}$	$\begin{bmatrix} 0.075 \\ 0.075 \\ 0.266 \end{bmatrix}$	$\begin{bmatrix} -0.075 \\ 0.075 \\ 0.266 \end{bmatrix}$	$\begin{bmatrix} -0.075 \\ -0.075 \\ 0.266 \end{bmatrix}$

Table 4.9: Geometric parameters of the IRMA-4U *UACDPR*

reported in Table 4.8, and the radius of each pulley is 0.03 m. For the wrench-feasible constant-orientation workspace computation (Sec. 4.3.2), the cable tension bounds are set to $\tau_{lb} = 4$ N and $\tau_{ub} = 100$ N, and the end-effector orientation is fixed to $\boldsymbol{\epsilon}_{const} = [0 \ 0 \ 0]^T$ deg. The resulting position bounds used in the GA pose-selection (Sec. 4.3.2) are $\mathbf{p}_{lb} = [0.1 \ -0.15 \ 0.5]^T$ m and $\mathbf{p}_{ub} = [0.45 \ 0.3 \ 1.2]^T$ m, chosen to maximize workspace coverage while maintaining a safety margin from workspace boundaries.

The *UACDPR* platform has mass $m = 12.992$ kg and center-of-mass offset $\mathbf{s}' = [0, 0, 0.173]^T$ m. The corresponding geometric parameters are reported in Table 4.9 and the radius of each pulley is 0.03 m. For the static workspace computation (Sec. 4.3.2), the tension bounds are set to $\tau_{lb} = 20$ N and $\tau_{ub} = 350$ N, yielding the position bounds $\mathbf{p}_{lb} = [-0.6, -0.6, 0.5]^T$ m and $\mathbf{p}_{ub} = [0.6, 0.6, 1.1]^T$ m.

The IRMA-4U prototype is equipped with incremental encoders on the motor winches (resolution 0.144°), incremental encoders on the swivel pulleys (resolution 0.03°), and an Xsens MTi-630 AHRS providing nominal accuracy 0.2° (roll/pitch) and 1.0° (yaw). The *OCDPR* prototype is equipped exclusively with motor incremental encoders for cable-length measurements (resolution 0.03°).

Although the actual sensor set of the *OCDPR* is limited, all sensor combinations considered in Sec. 4.3.3 are analyzed in simulation for both architectures to enable a comprehensive performance comparison. It is important to note that for the *UACDPR*, the self-calibration problem is overdetermined only when using at least the *LENGTH-SWIVEL* and *LENGTH-SWIVEL-AHRS* sensor sets. This is due to the fact that considering only *LENGTH* or *SWIVEL* data results in an underdetermined system of equations ($Nn < 6(N+1)$ for every N). Furthermore, while the combination of *LENGTH-AHRS* might algebraically appear to lead to an overdetermined system for a *UACDPR* with $n = 4$ cables, as shown by the limit:

$$\lim_{N \rightarrow \infty} \frac{7N}{6(N+1)} = \frac{7}{6} > 1 \quad (4.44)$$

the kinematic equations associated with the *AHRS* depend only on a subset of the

δ_p [m]	δ_e [deg]	σ_l^2 [m ²]	σ_σ^2 [deg ²]	σ_ϕ^2 [deg ²]	σ_θ^2 [deg ²]	σ_χ^2 [deg ²]
0.15	7.5	0.004^2	0.2^2	0.4^2	0.4^2	0.4^2

Table 4.10: Positioning error and sensor noise model parameters.

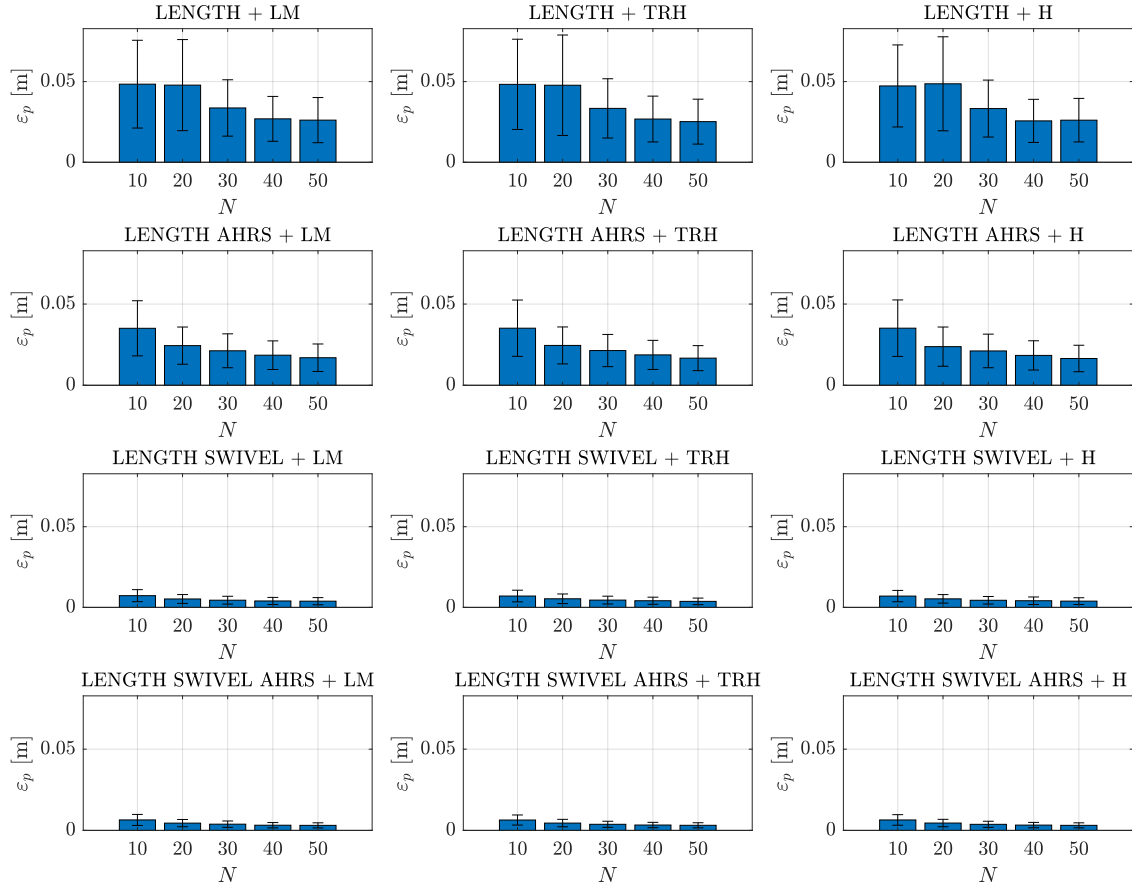
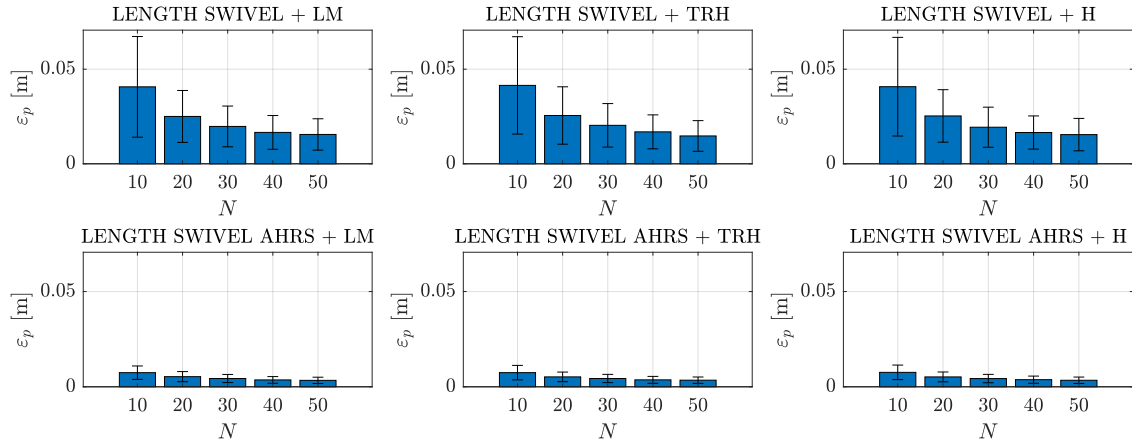
pose, specifically, the orientation. Consequently, the *LENGTH-AHRS* set is insufficient for initial pose calibration, which necessitates the inclusion of swivel angle measurements to resolve the full platform state for *UACDPRs*. Therefore, only the *LENGTH-SWIVEL* and *LENGTH-SWIVEL-AHRS* configurations are considered for this architecture. Based on the above specifications, and considering the contribution of modelling errors, identical values are adopted for the sensor noise variances in the simulations due to the comparable scale of the two robotic systems. For each sensor set S , the measurement-noise covariance is taken as diagonal, $Q_v = \text{diag}(\sigma_1^2, \dots, \sigma_m^2)$, with entries selected from Table 4.10 according to the included measurements (cable-length variations, swivel-angle variations, and AHRS components). Independent noise realizations are generated at each measurement pose. The bounds of pose-reaching errors δ_p and δ_e (Table 4.10) are selected to represent conservative open-loop control strategies, as typically employed during the data acquisition phase, when the actual *EE* pose is affected by significant uncertainty.

For each combination of the analyzed parameters, the number of Monte Carlo trials is set to $M = 1000$. This value is chosen after verifying that yields statistically stable estimates of the performance indices through the reaching of a minimum threshold on the relative error (in this case 2%) of the estimate in each performance index [122].

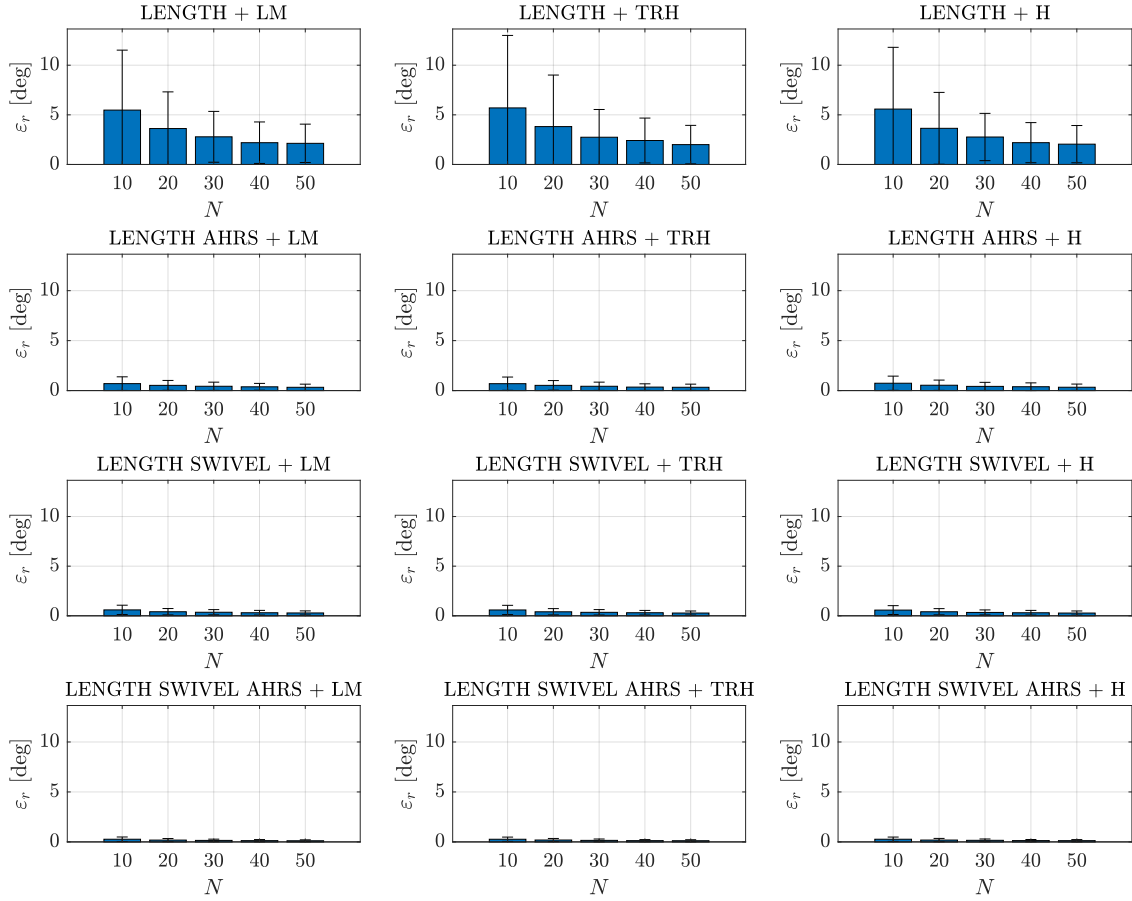
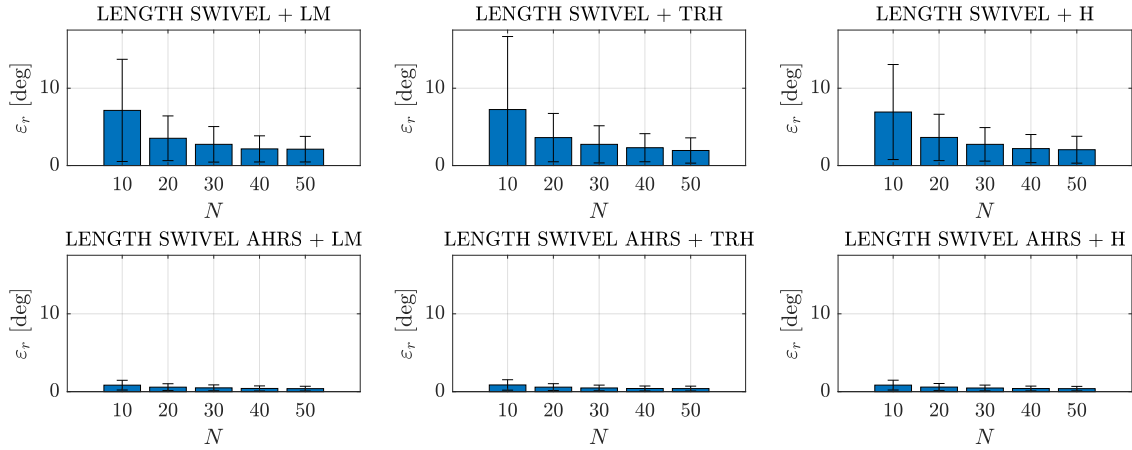
All simulations and experimental data processing are carried out in MATLAB R2023b on a PC equipped with an Intel i7-13650HX CPU at 2.60 GHz and 32 GB of RAM. The Levenberg–Marquardt (LM) and Trust-Region (TR) methods are implemented using the *lsqnonlin* and *fminunc* functions from the Optimization Toolbox, respectively, while Halley’s method is implemented following the update step detailed in Sec. 4.1. For *lsqnonlin*, the Levenberg–Marquardt algorithm is selected, while *fminunc* is used with the trust-region option. Analytic expressions of the cost function gradient and Hessian, derived in Sec. 4.1, are provided to the solvers. For all solvers, the standard MATLAB stopping criteria are adopted, with the tolerance on the cost function variation set to 10^{-9} , the tolerance on the solution variation to 10^{-7} , the tolerance on the gradient norm to 10^{-13} , and the maximum number of iterations limited to 100. The initial guess for the unknown parameter vector $\mathbf{x}_0$ is set equal to the ideal values used to generate the measurement poses.

The simulation results for the *OCDPR* prototype are summarized in Figs. 4.7, 4.9, 4.11, 4.13, 4.15, and 4.17, which report the statistics of the considered performance indices for all combinations of the analyzed parameters. For each performance index, the results are organized into a group of charts, where each chart corresponds to a specific combination of sensor set S and nonlinear solver K , and shows the mean value and standard deviation of the index as a function of the number of measurement poses N . Charts are arranged such that each row corresponds to a fixed sensor set and each column to a fixed solver, facilitating a direct comparison of the effects of S , K , and N . The same representation is adopted for the simulation results of the *UACDPR* prototype, reported in Figs. 4.8, 4.10, 4.12, 4.14, 4.16, and 4.18.

In agreement with previous analyses on initial-pose self-calibration [48, 60, 61, 63], the results show that increasing the number of measurement poses N and employing

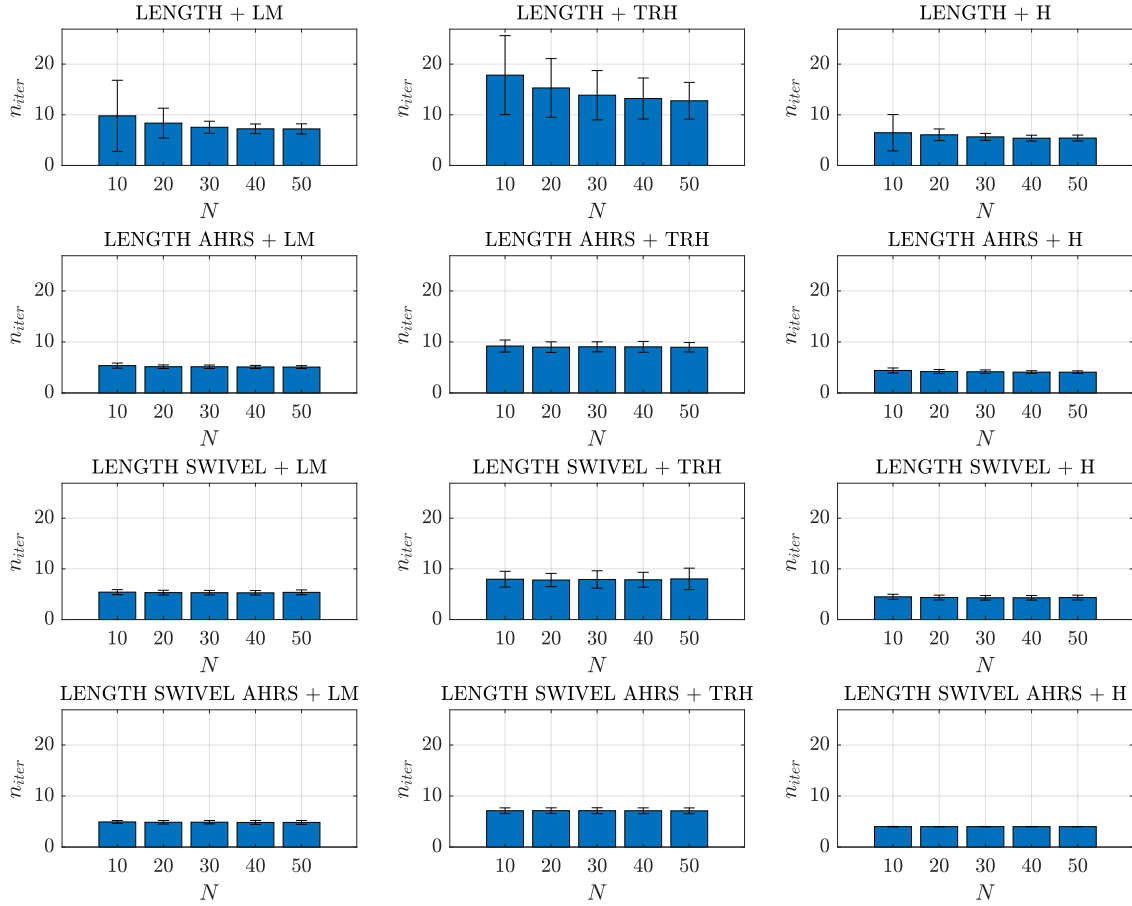
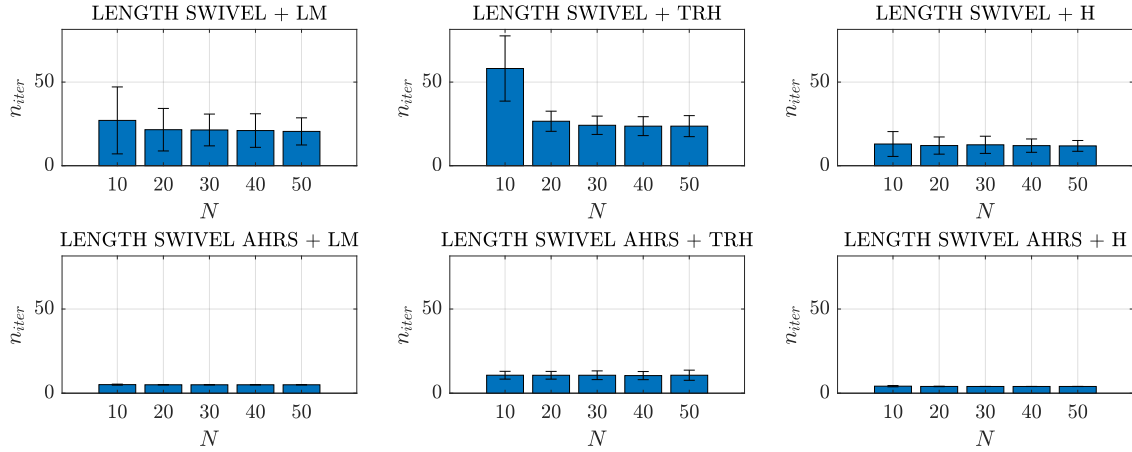

 Figure 4.7: Position error ε_p simulation results for the *OCDPR* prototype.

 Figure 4.8: Position error ε_p simulation results for the *UACDPR* prototype.

more informative sensor sets S lead to improved estimation accuracy. This trend is consistently observed as a reduction in both the position error ε_p and the orientation error ε_r for both architectures (Figs. 4.7–4.10). For the *OCDPR* prototype, augmenting cable-length measurements with swivel-angle encoders yields a marked improvement in position estimation accuracy, reducing ε_p by nearly one order of magnitude with respect to the *LENGTH* sensor set alone. The inclusion of *AHRS* measurements primarily improves orientation estimation, with a limited impact on position accuracy. A similar behavior is observed for the *UACDPR* prototype; however, due to the underactuated nature of the system, the availability of orientation measurements results in a substantially larger reduction of ε_r , leading to an order-of-magnitude improvement in

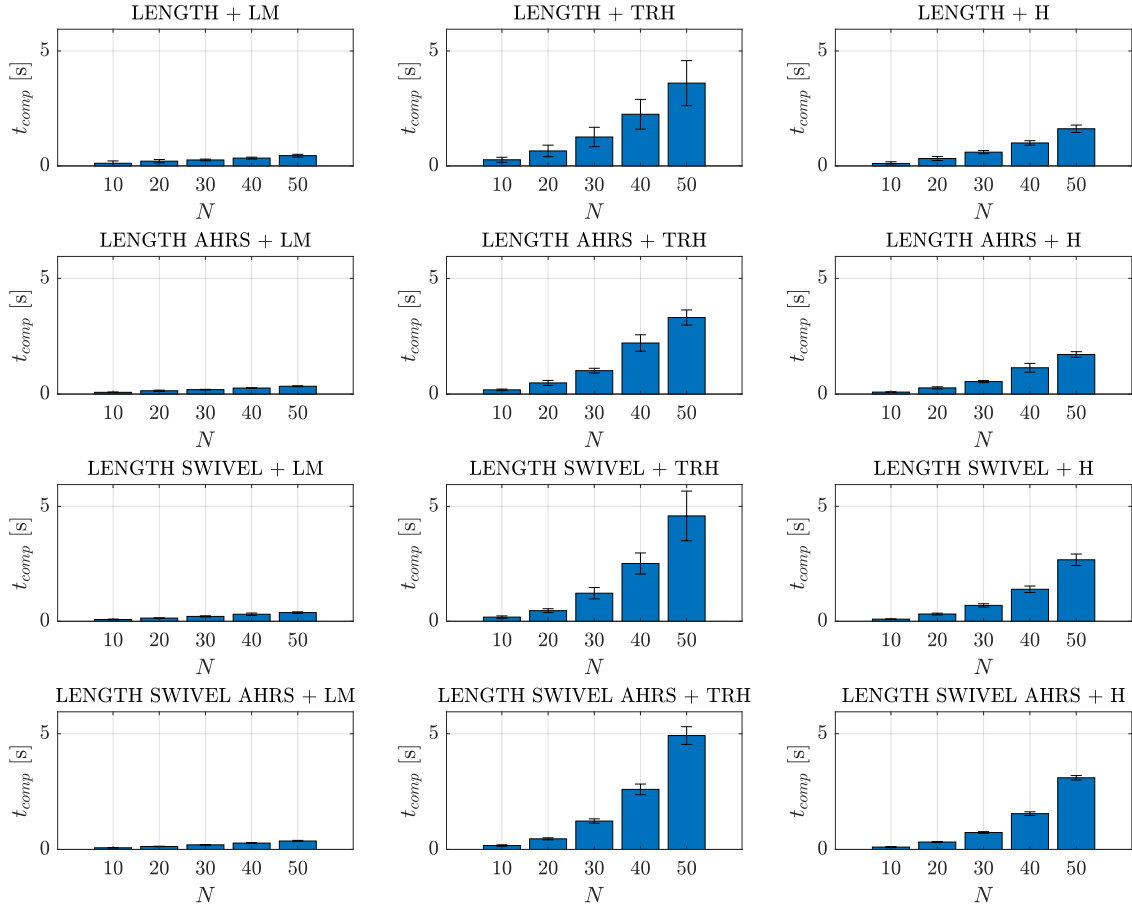
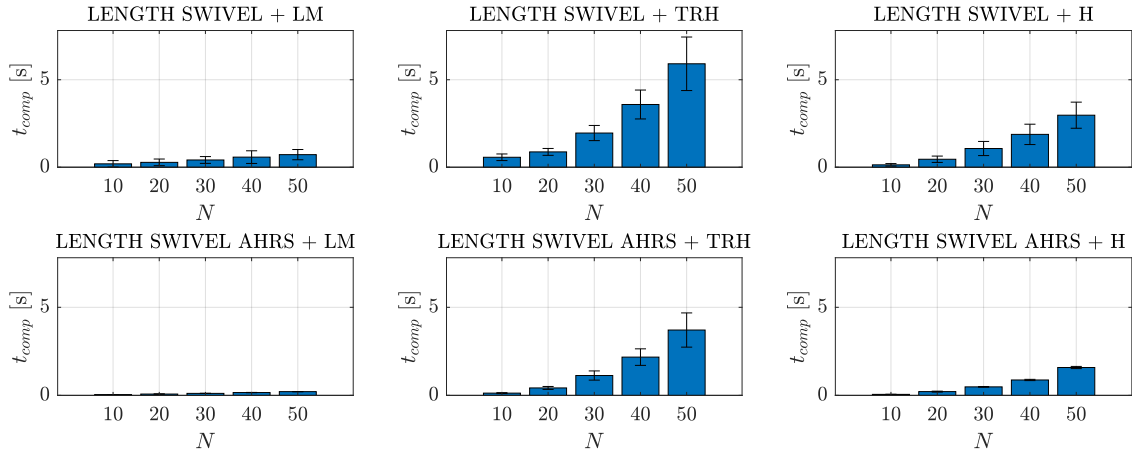
Figure 4.9: Orientation error ε_r simulation results for the *OCDPR* prototype.Figure 4.10: Orientation error ε_r simulation results for the *UACDPR* prototype.

orientation accuracy. Across all sensor sets and values of N , no substantial differences in estimation accuracy are observed among the three nonlinear solvers. The comparable mean errors and standard deviations indicate that, when convergence is achieved, all solvers consistently converge to the same solution, and that the choice of K has a negligible impact on the final estimation accuracy.

The number of iterations n_{iter} required for convergence remains overall stable for a fixed sensor set S , except when minimal sensor configurations are employed (i.e., the *LENGTH* set for the *OCDPR* and the *LENGTH-SWIVEL* set for the *UACDPR*), as shown in Figs. 4.11 and 4.12. In these cases, the reduced amount of information leads

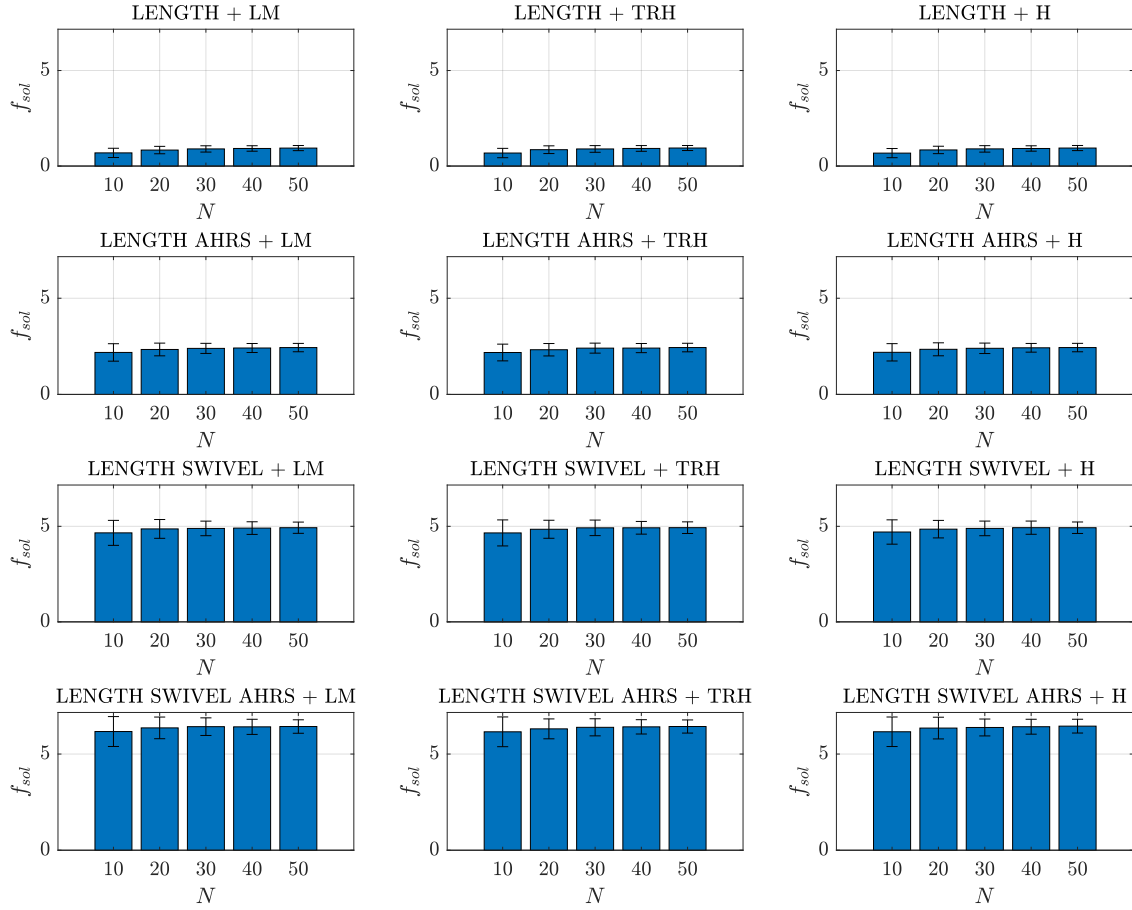
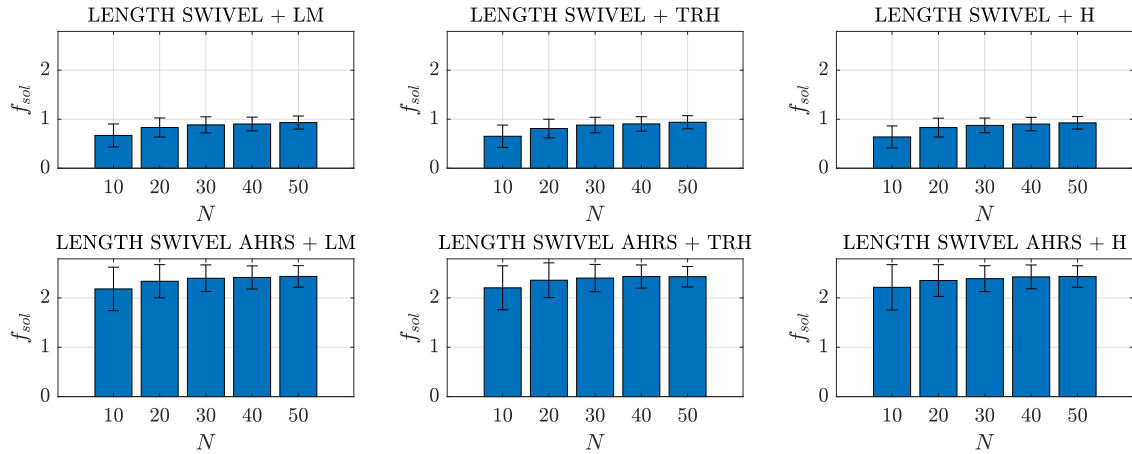

 Figure 4.11: Number of iterations n_{iter} simulation results for the *OCDPR* prototype.

 Figure 4.12: Number of iterations n_{iter} simulation results for the *UACDPR* prototype.

to higher iteration counts and increased variability across trials. Among the three solvers, the Trust-Region method generally requires the highest number of iterations, while Halley's method consistently exhibits the lowest iteration count, with Levenberg-Marquardt showing intermediate behavior. The computation time t_{comp} , reported in Figs. 4.13 and 4.14, increases with the number of measurement poses N for all sensor sets and solvers. An approximately linear scaling with N is observed for the Levenberg-Marquardt method, whereas both the Trust-Region and Halley solvers exhibit a super-linear growth, which becomes more pronounced as N increases. As a consequence of the additional second-order computations involved in these methods, the Levenberg-

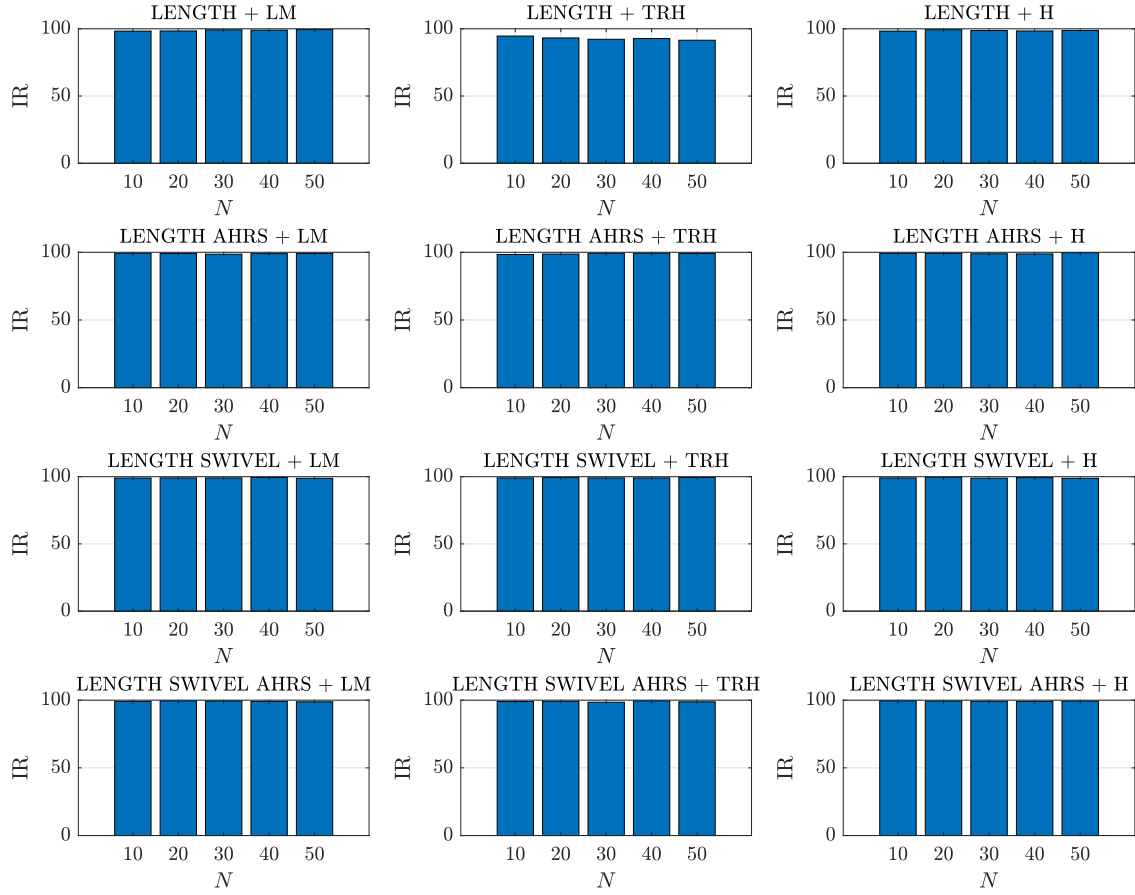
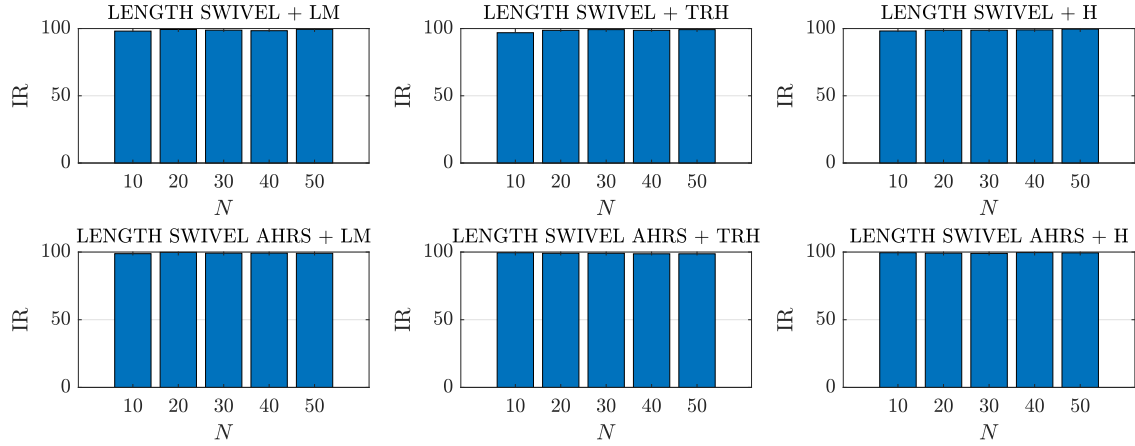
Figure 4.13: Computation time t_{comp} simulation results for the *OCDPR* prototype.Figure 4.14: Computation time t_{comp} simulation results for the *UACDPR* prototype.

Marquardt solver consistently achieves the lowest computation time, followed by Halley's method, while the Trust-Region solver results in the highest computational cost.

The final value of the cost function at convergence, f_{sol} , reported in Figs. 4.15 and 4.16, is identical across solvers for fixed values of N and S , confirming that all methods converge to the same minimum when convergence is achieved. Despite the normalization of the cost function with respect to the number of measurements, f_{sol} exhibits an increasing trend with the number of measurement poses N , particularly for less informative sensor sets. This reflects the accumulation of measurement noise which impacts more the cost function value when multiple sensors are used.

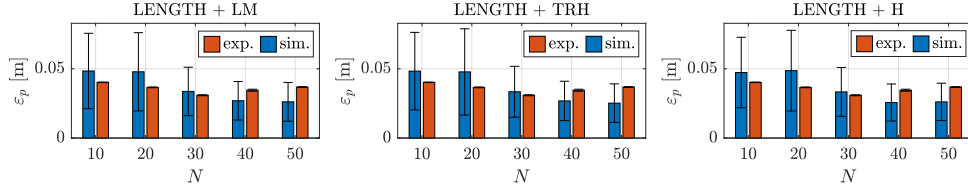
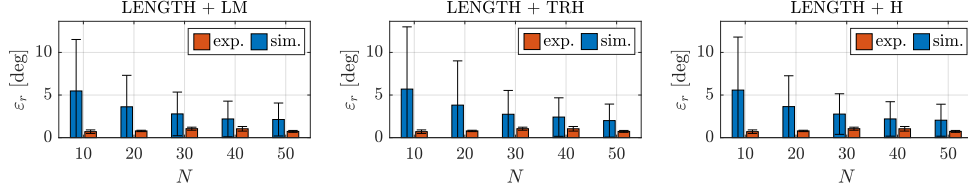

 Figure 4.15: Final cost function value f_{sol} simulation results for the *OCDPR* prototype.

 Figure 4.16: Final cost function value f_{sol} simulation results for the *UACDPR* prototype.

The inlier rate, reported in Figs. 4.17 and 4.18, remains close to 100% for both architectures across almost all configurations. A noticeable reduction is observed only for the *OCDPR* when using the minimal sensor set in combination with the Trust-Region solver. In all other cases, the high inlier rate indicates robust convergence of the self-calibration procedure.

Figure 4.17: Inlier rate simulation results for the *OCDPR* prototype.Figure 4.18: Inlier rate simulation results for the *UACDPR* prototype.

4.3.5 Experimental validation

For experimental validation of the proposed self-calibration method, self-calibration data were collected from both the *ARDC OCDPR* and the *IRMA-4U UACDPR* prototypes. A Vicon motion capture system was used to provide a ground-truth measurement of the end-effector pose, tracking the position of a set of reflective markers rigidly attached to the moving platform with an accuracy of ± 0.2 mm. The ground-truth end-effector pose was reconstructed from the measured marker positions using a geometry-based fitting procedure [123]. The self-calibration algorithm was then executed offline using the experimental data and the same implementation and param-

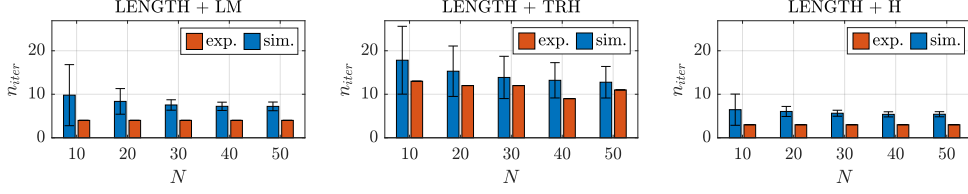
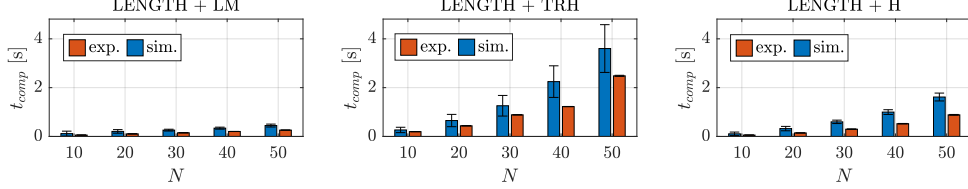
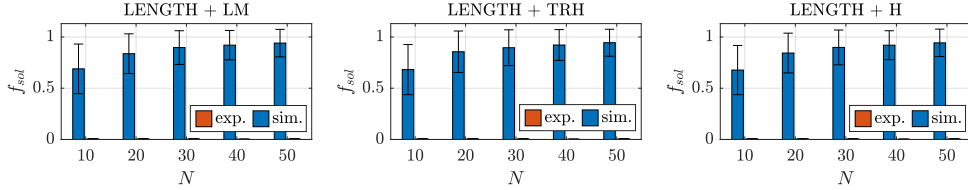
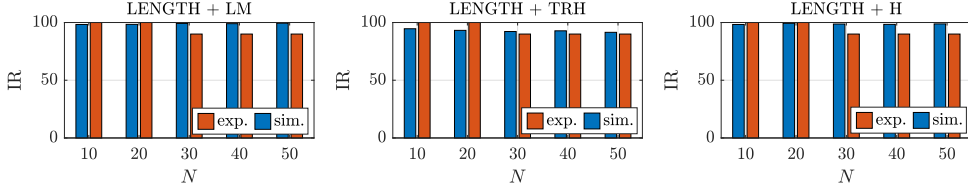

 Figure 4.19: Position error ε_p experiments results for the *OCDPR* prototype.

 Figure 4.20: Orientation error ε_r experiments results for the *OCDPR* prototype.

eter settings adopted in the simulation analysis. The pose error performance indices were finally obtained by comparing the estimated initial end-effector pose with the corresponding ground-truth measurement.

The number of experimental trials was set to $M = 10$ for both prototypes, providing a representative statistical sample while ensuring experimental repeatability and a reasonable total acquisition time. All remaining parameters, including the sensor configurations and the optimal measurement poses, were selected consistently with the simulation analysis.

For each trial, the *CDPR* was first moved to a pre-determined initial pose ζ_0 within the workspace. For both prototypes, an open-loop control strategy based on a rough estimate of the end-effector pose was used to reach the desired poses. Although the specific control law is not detailed here, as it is not the focus of this work, it is important to note that such open-loop strategies—typically adopted during this phase of *CDPR* operation—lead to non-negligible pose-reaching errors, consistently with the assumptions introduced in Sec. 4.3.3. At the initial pose, both the *CDPR* sensor measurements and the ground-truth end-effector pose were recorded. The robot was then commanded to sequentially reach each of the N measurement poses ζ_k , $k = 1, \dots, N$. At each pose, a stop phase was imposed to allow the system to settle, during which sensor measurements were acquired. In the data processing stage, measurements from each sensor channel were averaged over the stop phase to mitigate the effect of measurement noise. The resulting dataset was finally used to run the self-calibration algorithm offline, following the same procedure adopted in the simulation analysis.

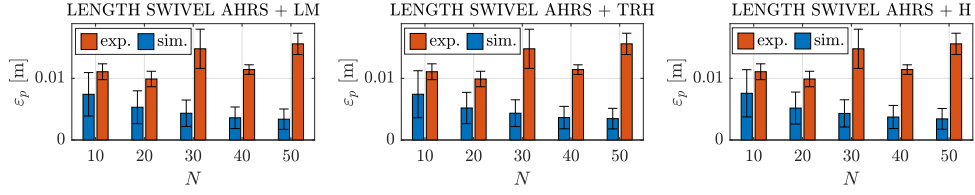
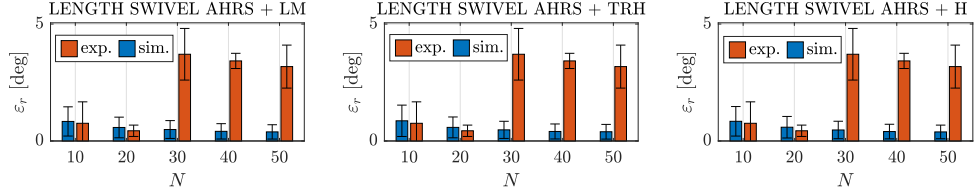
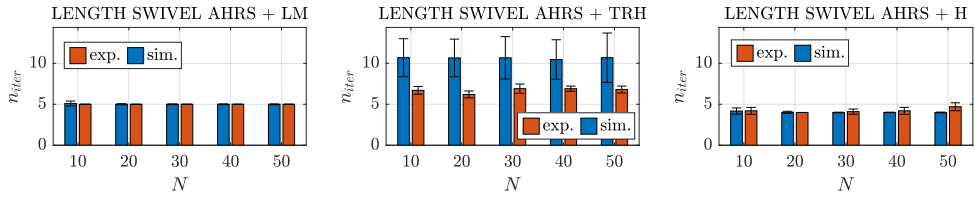
The experimental results for the *OCDPR* prototype are reported and compared with the corresponding simulation outcomes in Figs. 4.19–4.24. For this prototype, the sensor set S is limited to *LENGTH*, as cable-length measurements are the only sensing modality available. A consistent and prominent difference between experimental and simulated results is observed in the variability of all performance indices. In particular, the experimental standard deviation is significantly lower than the simulated one across all configurations. This discrepancy can be attributed to the different nature of pose-reaching errors in simulation and experiments. While the Monte Carlo simulations model pose-reaching errors as independent random perturbations at each measurement pose, the experimental procedure is affected by repeatable, trial-consistent systematic errors, such as calibration offsets and repeatable control biases. As a consequence, the experimental measurement poses exhibit reduced variability across trials,

Figure 4.21: Number of iterations n_{iter} experiments results for the *OCDPR* prototype.Figure 4.22: Computation time t_{comp} experiments results for the *OCDPR* prototype.Figure 4.23: Final cost function value f_{sol} experiments results for the *OCDPR* prototype.Figure 4.24: Inlier rate experiments results for the *OCDPR* prototype.

leading to smaller dispersion of the estimated performance indices. The experimental position error ε_p does not decrease monotonically with the number of measurement poses N , as observed in simulation. Instead, it reaches a minimum for intermediate values of N (approximately $N = 20$) and slightly increases for larger N . This behavior is consistent with the presence of systematic pose-reaching errors, which introduce biased measurements that do not average out when additional poses are included, thus limiting the achievable accuracy beyond a certain number of measurements. The orientation error ε_r remains approximately constant with respect to N and is significantly lower in the experiments than in the simulations. This result suggests that the orientation of the moving platform is less affected by pose-reaching uncertainties in the experimental setup. The number of iterations n_{iter} and the computation time t_{comp} exhibit trends consistent with the simulation results, while attaining lower values in the experiments. This behavior is likely due to the reduced errors in the experimental measurements, which facilitates faster and more stable convergence of the nonlinear solvers.

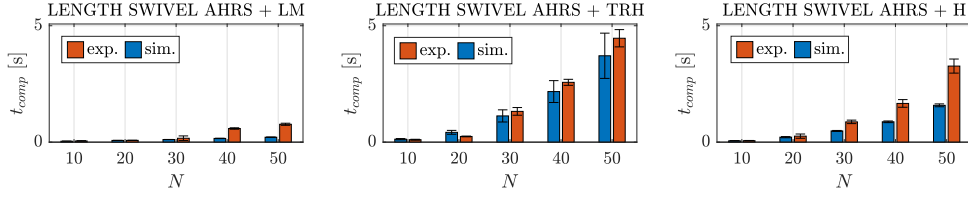
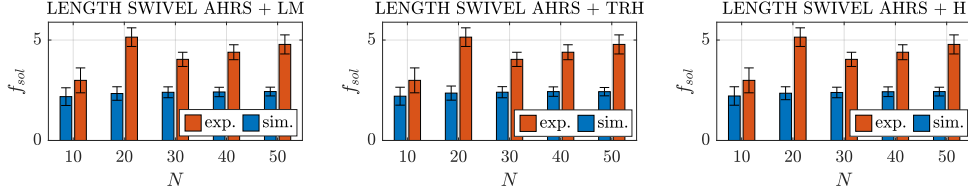
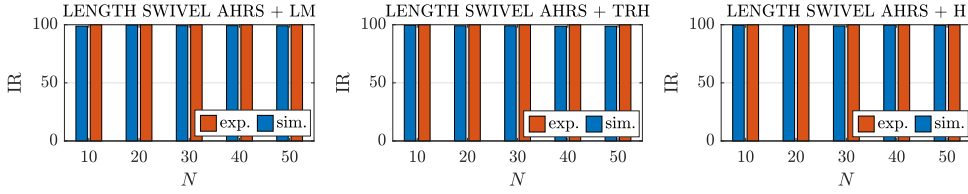
Similarly, the final cost function value at convergence f_{sol} is substantially lower in the experiments than in the simulations, reflecting the lower dispersion of the residuals. Finally, the inlier rate remains high across all configurations, although a slight decrease with increasing N is observed in the experimental results.

The experimental results for the *UACDPR* prototype are reported and compared


 Figure 4.25: Position error ε_p experiments results for the *UACDPR* prototype.

 Figure 4.26: Orientation error ε_r experiments results for the *UACDPR* prototype.

 Figure 4.27: Number of iterations n_{iter} experiments results for the *UACDPR* prototype.

with the corresponding simulation outcomes in Figs. 4.25–4.30. In contrast to the *OCDPR* case, the experimental standard deviation of the performance indices is of the same order of magnitude as the simulated one. This indicates that, for the *UACDPR*, trial-to-trial variability is not dominated by repeatable systematic effects, probably due to the unconstrained nature of these systems, which makes them less repeatable. With respect to estimation accuracy, for $N < 30$ the experimental results follow the simulation trends, although with a slightly larger position error. For $N \geq 30$, instead, both the mean position and orientation errors are significantly larger with respect to the simulation. This discrepancy is likely due to unmodeled effects, such as swivel-pulley friction and the violation of the planarity constraint between the cable and the swivel pulley. This phenomenon is particularly pronounced under low cable-tension conditions. As previously established, swivel angle measurements and their associated kinematic equations are indispensable for ensuring that the initial-pose self-calibration remains an overdetermined problem for *UACDPRs*. Consequently, inaccuracies or modeling discrepancies in the swivel measurement model, as formulated in Eq. (2.9), can significantly degrade the final calibration results⁸. Despite the observed gap between experimental and simulated errors, the achieved accuracy is consistent with previous experimental self-calibration results on the same prototype [61], where a position error of 23 mm and an orientation error of 5.44° were obtained with the same sensor configuration and $N = 20$, but without optimal measurement-pose selection. Regarding computational performance, the number of iterations n_{iter} and the computation time t_{comp} exhibit trends comparable to simulation for the Levenberg–Marquardt and Halley solvers. Conversely, the Trust-Region solver attains smaller iteration counts while requiring comparable computation time in the experiments. This behavior suggests an earlier termination of the optimization process, likely caused by the presence of higher

⁸A specific and robust analysis of the relationship between swivel angle measurement accuracy and initial-pose self-calibration will be the subject of future studies.

Figure 4.28: Computation time t_{comp} experiments results for the *UACDPR* prototype.Figure 4.29: Final cost function value f_{sol} experiments results for the *UACDPR* prototype.Figure 4.30: Inlier rate experiments results for the *UACDPR* prototype.

residual floors induced by unmodeled effects, rather than an improved convergence behavior. The final cost function value at convergence, f_{sol} , is consistently higher in the experiments than in the simulations, further confirming the presence of modeling discrepancies and non-ideal measurement conditions. Finally, the experimental inlier rate is consistent with the simulated one.

Overall, the experimental results show that the proposed framework for parameter and algorithm analysis is effective in predicting the actual performance trends of initial-pose self-calibration on real *CDPR* systems. This is particularly true for the *OCDPR* architecture, where redundant actuation enables the use of more informative measurements, such as cable-length variations from motor encoders, resulting in increased robustness to pose-reaching uncertainties and unmodeled disturbances. From a practical perspective, the sensor set S is the most influential design choice, with swivel-angle measurements providing a substantial improvement in position accuracy and *AHRS* measurements being essential for reliable orientation estimation, especially in underactuated architectures. Increasing the number of measurement poses N improves performance up to a saturation point, beyond which experimental accuracy may degrade due to systematic errors; in the analyzed setups, values of N between 20 and 30 represent a favorable trade-off. When convergence is achieved, the choice of the nonlinear solver K has a limited impact on estimation accuracy and mainly affects convergence speed and computational cost, with the Levenberg–Marquardt method offering the most balanced performance. At the same time, the results highlight intrinsic limitations of the self-calibration procedure, as unmodeled systematic effects and modeling inaccuracies ultimately bound the achievable accuracy, particularly for underactuated architectures. As a result, while the proposed framework provides effective guidance for parameter and algorithm selection, the final performance is still mainly influenced by modelling errors and sensing quality.

4.4 Online initial-pose self-calibration

While the procedures described in Sections 4.2 and 4.3 provide robust solutions for system initialization, they rely on sequential offline batch processing, wherein data acquisition must be entirely completed before optimization begins. To further reduce the overall time required for the robot to reach operational status, this section introduces novel online self-calibration techniques. These methods aim to estimate the initial end-effector pose ζ_0 concurrently with the state estimation of the current pose ζ_k during the execution of the robot's initial trajectory. This approach streamlines the transition from power-on to full operational readiness. The CDPR is assumed to be equipped with the internal sensor suite described in Section 4.2.2, which includes: winch motor encoders for cable-length variations $\Delta \mathbf{l}_k$, swivel-pulley encoders for swivel-angle variations $\Delta \boldsymbol{\sigma}_k$, and an AHRS rigidly attached to the platform to provide absolute roll/pitch (ϕ_k, θ_k) and relative yaw variations $\Delta \chi$. Alternatively, raw IMU data, such as acceleration $\ddot{\mathbf{p}}_{IMU}$ and gyroscope outputs $\boldsymbol{\omega}_{IMU}$, may be utilized.

Various methodologies are explored for performing online self-calibration. Given that the optimization problem formulated in Sec. 4.1 is rooted in least squares, a modified Recursive Least Squares (RLS) algorithm is presented in Sec. 4.4.2. Furthermore, assuming the availability of an IMU or an AHRS for local inertial data, an Extended Kalman Filter (EKF) is proposed in Sec. 4.4.3. Because these methods rely on the linearization of kinematic equations, they may incur significant estimation errors when model nonlinearities are pronounced. To address this, *unscented* versions of both the RLS and the EKF are provided in Sections 4.4.2.1 and 4.4.3.1, respectively. An experimental assessment is conducted for the IRMA-4U UACDPR introduced in Sec. 4.3.5. Finally, the performance of these novel online techniques is compared against the state-of-the-art *nested* method introduced by Caverly et al. [58, 59], detailed in Sec. 4.4.1.

4.4.1 Nested method

In the nested approach, the self-calibration problem is partitioned into two distinct timescales: an inner loop and an outer loop. The inner loop is responsible for high-frequency state estimation, determining the current EE pose ζ_k whenever new sensor data becomes available. This estimation relies on the most recent estimate of the initial pose $\hat{\zeta}_0$ provided by the calibration loop. Conversely, the outer loop operates at a lower frequency, solving the initial-pose self-calibration problem only when a batch of N measurements with sufficient information has been collected.

The inner loop implementation follows the pose estimation framework described in [50], formulated as a NWLS optimization:

$$\hat{\zeta}_k = \arg \min_{\zeta_k} \frac{1}{2} \check{\mathbf{r}}_k^T \check{\mathbf{W}} \check{\mathbf{r}}_k, \quad (4.45)$$

where the residual vector $\check{\mathbf{r}}_k$ is a function of the current pose ζ_k and the reference values derived from the last estimate of the initial pose $\hat{\zeta}_0$:

$$\check{\mathbf{r}}_k = \begin{bmatrix} \mathbf{l}(\zeta_k) - \mathbf{l}(\hat{\zeta}_0) - \Delta \mathbf{l}_k^* \\ \boldsymbol{\sigma}(\zeta_k) - \boldsymbol{\sigma}(\hat{\zeta}_0) - \Delta \boldsymbol{\sigma}_k^* \\ \phi_k - \phi_k^* \\ \theta_k - \theta_k^* \\ \chi(\zeta_k) - \chi(\hat{\zeta}_0) - \Delta \chi_k^* \end{bmatrix}. \quad (4.46)$$

The weighting matrix $\check{\mathbf{W}}$ is computed as the inverse of the measurement noise covariance, consistent with Eq. (4.5).

The outer loop solves a calibration problem structurally identical to the offline NWLS formulation presented in Sec. 4.1. To ensure the robustness of this estimate, the dataset is managed through a sliding buffer of N measurements. After accumulating the first N measurement time-steps, the indices for the data buffer are initialized as $k_1 = 1, k_2 = 2, \dots, k_N = N$. As subsequent measurements are recorded, the dataset may be updated to incorporate more informative configurations. The criteria for updating this buffer are based on two observability metrics: the *Position Dilution of Precision* (*PDOP*) and the *Orientation Dilution of Precision* (*ODOP*). These metrics are derived from the covariance matrix $\mathbf{P}$ of the unknown vector, obtained by linearizing the system at the current estimate:

$$\mathbf{P} = (\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} = \begin{bmatrix} \mathbf{P}_{\zeta_0 \zeta_0} & \mathbf{P}_{\zeta_0 \zeta} \\ \mathbf{P}_{\zeta \zeta_0} & \mathbf{P}_{\zeta \zeta} \end{bmatrix}, \quad (4.47)$$

where $\mathbf{P}_{\zeta \zeta} \in \mathbb{R}^{6N \times 6N}$ represents the covariance of the N measurement poses. By extracting the k_i -th 6×6 diagonal block of $\mathbf{P}_{\zeta \zeta}$, denoted by $\mathbf{P}_{\zeta}^{(k_i)}$, which represents the covariance of the k_i -th pose ζ_{k_i} , and letting $\mathbf{P}_{\zeta, p}^{(k_i)}$ denote the upper-left 3×3 subblock of $\mathbf{P}_{\zeta}^{(k_i)}$ (position covariance), and $\mathbf{P}_{\zeta, r}^{(k_i)}$ denote the bottom-right 3×3 subblock (orientation covariance), we can define the observability indices as:

$$\text{PDOP} = \sqrt{\frac{1}{N} \sum_{k_i=1}^N \text{trace}(\mathbf{P}_{\zeta, p}^{(k_i)})}, \quad \text{ODOP} = \sqrt{\frac{1}{N} \sum_{k_i=1}^N \text{trace}(\mathbf{P}_{\zeta, r}^{(k_i)})}. \quad (4.48)$$

The practical meaning of the covariance matrix $\mathbf{P}$ lies in its characterization of the statistical "dispersion" of the estimated state. It essentially conveys the confidence level of the estimation; a high dispersion implies low precision, whereas a concentrated covariance indicates that the estimate is tightly clustered around the true state. In this context, *PDOP* and *ODOP* effectively quantify the precision of the positional and rotational estimates, respectively. These indices directly correspond to the standard deviation of the position and orientation errors, respectively. Once the buffer is full, new measurements are tentatively swapped with randomly selected entries. The *PDOP* and *ODOP* are recomputed; if the swap results in a decrease of these indices—indicating improved precision—and they fall below a predefined threshold, the entry is updated and the outer loop is executed. Otherwise, the data is discarded. The complete flow is illustrated in Fig. 4.31.

4.4.2 Recursive Least Squares

While the batch formulation introduced in Sec. 4.1 is effective for offline analysis, the Recursive Least Squares (*RLS*) algorithm provides a computationally efficient framework for online estimation. *RLS* serves as the recursive counterpart to the *NWLS* problem, allowing the parameter vector $\mathbf{x}$ to be updated incrementally as each new measurement is sampled. In its standard form, *RLS* is suitable for estimating a constant parameter vector from a sequence of measurements that fit the linear model [124]:

$$\mathbf{y}_k = \mathbf{S}_k \mathbf{x} + \mathbf{v}_k, \quad (4.49)$$

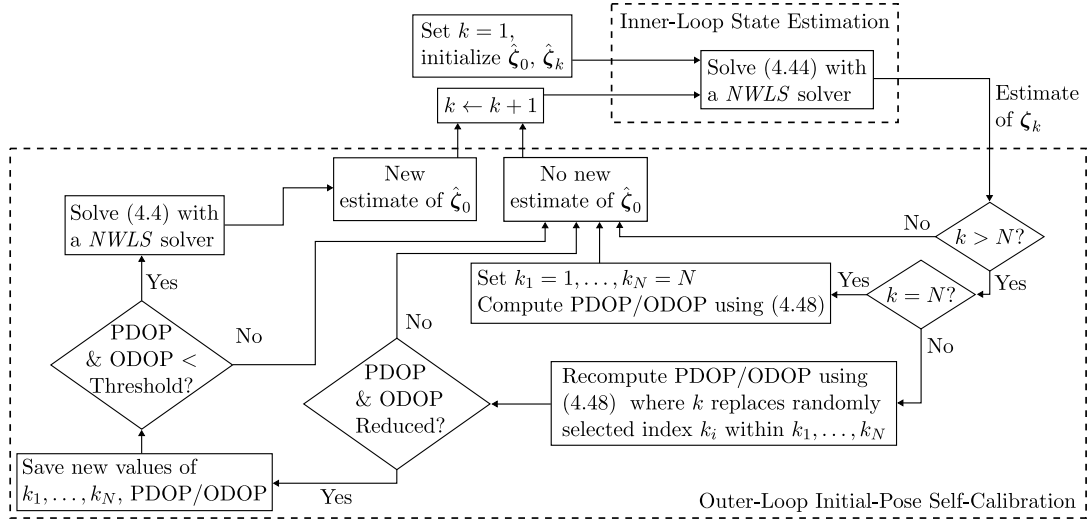


Figure 4.31: Flowchart of the nested online initial-pose self-calibration algorithm.

where $\mathbf{y}_k$ is the measurement vector at step k , $\mathbf{S}_k$ is called the observation matrix, and $\mathbf{v}_k$ is the measurement noise (generally assumed as zero mean and gaussian $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}_{v,k})$).

The filter is initialized with an estimate of the parameter vector $\hat{\mathbf{x}}_0$ and its associated uncertainty covariance $\mathbf{P}_0$:

$$\hat{\mathbf{x}}_0 = E(\mathbf{x}), \quad \text{and} \quad \mathbf{P}_0 = E[(\mathbf{x} - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]. \quad (4.50)$$

As measurements are sampled for $k = 1, 2, \dots$, the estimate is refined by iterating through the following update equations:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k-1} \mathbf{S}_k^T (\mathbf{S}_k \mathbf{P}_{k-1} \mathbf{S}_k^T + \mathbf{Q}_{v,k})^{-1}, \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{S}_k \hat{\mathbf{x}}_{k-1}), \\ \mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{P}_{k-1}, \end{aligned} \quad (4.51)$$

where $\hat{\mathbf{x}}_k$ is the estimation of the parameter vector at time k , $\mathbf{P}_k$ is its covariance matrix, $\mathbf{K}_k$ is the estimator gain matrix, and $\mathbf{Q}_{v,k}$ is the covariance matrix of the measurement noise.

Setting $\mathbf{x} = \boldsymbol{\zeta}_0$, and the measurement vector $\mathbf{y}_k$ as:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{l}(\hat{\boldsymbol{\zeta}}_k) - \Delta \mathbf{l}_k^* \\ \boldsymbol{\sigma}(\hat{\boldsymbol{\zeta}}_k) - \Delta \boldsymbol{\sigma}_k^* \\ \phi_k^* \\ \theta_k^* \\ \chi(\hat{\boldsymbol{\zeta}}_0) - \Delta \chi_k^* \end{bmatrix} = \begin{bmatrix} \mathbf{l}(\boldsymbol{\zeta}_0) \\ \boldsymbol{\sigma}(\boldsymbol{\zeta}_0) \\ 0 \\ 0 \\ \chi(\boldsymbol{\zeta}_0) \end{bmatrix} + \begin{bmatrix} v_l \\ v_\sigma \\ v_\phi \\ v_\theta \\ v_\chi \end{bmatrix}, \quad (4.52)$$

where $\hat{\boldsymbol{\zeta}}_k$ is the last available estimation of the *EE* pose, the formulation 4.49 can be obtained linearizing the geometric constraints (Eqs. (2.13), (2.9)) around the current estimate $\mathbf{x}^9$. However, this approach decouples state estimation from parameter calibration. By treating $\boldsymbol{\zeta}_k$ merely as an external input, the filter ignores the inherent coupling and the uncertainty of the current pose, which often leads to poor convergence.

⁹Please note that if instead of considering the initial *EE* pose, we consider the initial reference values of the measured quantities as unknown parameters $\mathbf{x} = [\mathbf{l}_0^T \boldsymbol{\sigma}_0^T \chi_0]^T$, the relation is already linear.

To account for these interactions, a modified *RLS* approach is proposed using an augmented state vector $\mathbf{x}_k = [\zeta_0^T \zeta_k^T]^T$, which simultaneously includes the constant initial pose and the time-varying current pose. Since ζ_k changes at every time-step, the covariance matrix blocks associated with the current pose must be reassigned at each iteration to reflect the new state guess. The measurement vector for this augmented system is expressed as:

$$\mathbf{y}_k = \begin{bmatrix} \Delta \mathbf{l}_k^* \\ \Delta \boldsymbol{\sigma}_k^* \\ \phi_k^* \\ \theta_k^* \\ \Delta \chi_k^* \end{bmatrix} = \begin{bmatrix} \mathbf{I}(\zeta_k) \\ \boldsymbol{\sigma}(\zeta_k) \\ \phi_k \\ \theta_k \\ \chi(\zeta_k) \end{bmatrix} - \begin{bmatrix} \mathbf{I}(\zeta_0) \\ \boldsymbol{\sigma}(\zeta_0) \\ 0 \\ 0 \\ \chi(\zeta_0) \end{bmatrix} + \mathbf{v}_k, \quad (4.53)$$

with the observation matrix $\mathbf{S}_k$ obtained by linearizing Eq. (4.53) around the augmented state $\mathbf{x}_k$:

$$\mathbf{S}_k = \begin{bmatrix} -\mathbf{J}_{l,k}(\zeta_0) & \mathbf{J}_{l,k}(\zeta_k) \\ -\mathbf{J}_{\sigma,k}(\zeta_0) & -\mathbf{J}_{\sigma,k}(\zeta_k) \\ -\mathbf{J}_{\chi,k} & \mathbf{J}_{\epsilon,k} \end{bmatrix}. \quad (4.54)$$

Given the time-varying nature of ζ_k , the recursive update is modified to incorporate an *a priori* state and covariance reset:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \begin{bmatrix} \hat{\zeta}_{0,k-1}^+ \\ \hat{\zeta}_{k,guess} \end{bmatrix}, \\ \mathbf{P}_k^- &= \begin{bmatrix} \mathbf{P}_{\zeta_0,k-1}^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\zeta_{k,guess}} \end{bmatrix}, \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{S}_k^T (\mathbf{S}_k \mathbf{P}_k^- \mathbf{S}_k^T + \mathbf{Q}_{v,k})^{-1}, \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)), \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{S}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{S}_k)^T + \mathbf{K}_k \mathbf{Q}_{v,k} \mathbf{K}_k^T, \end{aligned} \quad (4.55)$$

where $\hat{\mathbf{x}}_k^-$ is the *a priori* estimate of the unknowns, $\hat{\zeta}_{0,k-1}^+$ is the *a posteriori* estimate of the initial pose and $\mathbf{P}_{\zeta_0,k-1}^+$ is the corresponding block of the covariance matrix, $\hat{\zeta}_{k,guess}$ is a guess of the current pose, and $\mathbf{P}_{\zeta_{k,guess}}$ is a guess of the corresponding covariance matrix block. The initial guess $\hat{\zeta}_{k,guess}$ is set to the current control setpoint, assuming the system remains in sufficient proximity to the desired trajectory. The associated uncertainty, $\mathbf{P}_{\zeta_{k,guess}}$, is modeled as a diagonal matrix scaled by a constant tuning parameter α . This parameter is determined through empirical trial and error to ensure robust convergence. By adjusting α , the algorithm effectively balances the influence of the prior setpoint against new measurement data, providing a necessary regularization for the iterative update in this preliminary phase.

A limitation of this modified approach is that the resulting covariance matrix $\mathbf{P}_k$ is no longer fully representative of the cumulative uncertainty across the entire trajectory, as the state blocks are reset at each step. Consequently, if *PDOP* or *ODOP* indices are required to determine a termination condition, they must be computed independently of the *RLS* covariance to ensure they accurately reflect the global observability of the calibration procedure. Nevertheless, to evaluate the filter's performance within this preliminary study, *PDOP* and *ODOP* are still derived from the diagonal elements of the *a posteriori* covariance matrix $\mathbf{P}_k^+$, acknowledging the aforementioned limitations of this approach.

4.4.2.1 Unscented version

Instead of linearizing the nonlinear model to obtain the observation matrix $\mathbf{S}_k$, the unscented version of the RLS utilizes the *Unscented Transform (UT)*. This approach propagates a set of deterministically chosen sample points, known as sigma points, through the nonlinear model to capture the mean and covariance of the state distribution more accurately than first-order Taylor expansions [124].

Given a vector $\mathbf{x} \in \mathbb{R}^{n_x}$ with known mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$, and a nonlinear function $\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, the UT approximates the statistics of $\mathbf{y}$ through the following steps:

1. Generate sigma points:

$$\begin{aligned}\mathbf{x}^{(i)} &= \hat{\mathbf{x}} + \bar{\mathbf{x}}^i, \quad i = 1, \dots, 2n_x \\ \bar{\mathbf{x}}^{(i)} &= \left(\sqrt{n\mathbf{P}} \right)_i^T, \quad i = 1, \dots, n_x \\ \bar{\mathbf{x}}^{(n+i)} &= - \left(\sqrt{n\mathbf{P}} \right)_i^T, \quad i = 1, \dots, n_x.\end{aligned}\tag{4.56}$$

where $\sqrt{n_x\mathbf{P}}$ is the matrix square root such that $(\sqrt{n_x\mathbf{P}})^T \sqrt{n_x\mathbf{P}} = n_x\mathbf{P}$, and $(\sqrt{n_x\mathbf{P}})_i$ denotes the i -th row of the result.

2. Propagate sigma points:

$$\mathbf{y}^{(i)} = \mathbf{h}(\mathbf{x}^{(i)}), \quad i = 1, \dots, 2n_x\tag{4.57}$$

3. Compute predicted mean and covariance:

$$\hat{\mathbf{y}} = \frac{1}{2n_x} \sum_{i=1}^{2n_x} \mathbf{y}^{(i)}, \quad \mathbf{P}_y = \frac{1}{2n_x} \sum_{i=1}^{2n_x} (\mathbf{y}^{(i)} - \hat{\mathbf{y}})(\mathbf{y}^{(i)} - \hat{\mathbf{y}})^T + \mathbf{Q}_{v,k}\tag{4.58}$$

where $\mathbf{Q}_{v,k}$ is the covariance matrix of the measurement noise $\mathbf{v}_k$.

These steps are hereafter denoted by the operator: $\hat{\mathbf{y}}, \mathbf{P}_y = UT(\mathbf{h}(\cdot), \hat{\mathbf{x}}, \mathbf{P})$.

Considering the formulation of the self-calibration problem being the same as the RLS case, at each iteration k , the steps 4.55 are modified as follows:

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \begin{bmatrix} \hat{\boldsymbol{\zeta}}_{0,k-1}^+ \\ \hat{\boldsymbol{\zeta}}_{k,guess} \end{bmatrix}, \\ \mathbf{P}_k^- &= \begin{bmatrix} \mathbf{P}_{\zeta_0,k-1}^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\zeta_k,guess} \end{bmatrix}, \\ \hat{\mathbf{y}}_k, \mathbf{P}_y &= UT(\mathbf{h}(\cdot), \hat{\mathbf{x}}_k^-, \mathbf{P}_k^-), \\ \mathbf{P}_{xy} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} (\mathbf{x}_k^{(i)} - \hat{\mathbf{x}}_k^-)(\mathbf{y}_k^{(i)} - \hat{\mathbf{y}}_k)^T, \\ \mathbf{K}_k &= \mathbf{P}_{xy} \mathbf{P}_y^{-1}, \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k), \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T,\end{aligned}\tag{4.59}$$

where, as before, the state and covariance are initialized using the control setpoint for the iterative guess $\hat{\boldsymbol{\zeta}}_{k,guess}$ and a diagonal matrix $\mathbf{P}_{\zeta_k,guess} = \alpha \mathbf{I}$ as the prior uncertainty.

The primary advantage of the unscented approach is that the covariance block $\mathbf{P}_{\zeta_0,k}$ corresponding to the calibration parameters is more representative of the true uncertainty, leading to estimates of ζ_0 that are typically closer to the ground truth. However, as noted previously, the covariance matrix $\mathbf{P}_k$ still does not fully capture the global uncertainty of the measurement pose due to the iterative resetting of state blocks. Consequently, while *PDOP* and *ODOP* indices are derived from the diagonal of $\mathbf{P}_k^+$ for performance evaluation in this study, they remain independent of the global observability required for termination criteria. Finally, the improved fidelity comes at a higher computational cost, as the number of sigma points scales with the dimension of the state vector, potentially limiting real-time applicability in high-dimensional systems.

4.4.3 Extended Kalman Filter

In the past, several studies have demonstrated the efficacy of Kalman Filters (*KF*) for pose estimation of *CDPRs* [125–128]. The underlying idea behind these approaches is that including a dynamic model, rather than relying solely on the geometric relations, significantly improves estimation. Especially when an *IMU* is available for acceleration and rate of turn measurements of the moving platform ($\ddot{\mathbf{p}}_{IMU}$ and $\ddot{\omega}_{IMU}$), and thus, there is no need for inertial parameter identification (that can introduce further modeling errors) since no *CDPR* dynamics is used.

However, Kalman filtering can be used not only to estimate the states of a system, but also to estimate its unknown parameters [124]. The general discrete-time Extended Kalman Filter (*EKF*) can be applied to nonlinear systems expressed in the following form:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k),\end{aligned}\tag{4.60}$$

where $\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$ is the process model, $\mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k)$ is the observation model, $\mathbf{w}_k$ is the process noise, $\mathbf{v}_k$ is the measurement noise (generally assumed as zero mean and gaussian $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{w,k})$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{v,k})$), $\mathbf{u}_k$ is the input vector, $\mathbf{y}_k$ is the measurement vector, and $\mathbf{x}_k$ is the state vector. Generally, $\mathbf{f}_k$ is the dynamic model of the system and $\mathbf{h}_k$ is the kinematic model that relates the measurements to the system state. In the case of parameter estimation, the state $\mathbf{x}_k$ is augmented with the unknown parameters $\boldsymbol{\beta}_k$, and their update is modeled as:

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \mathbf{w}_{\beta,k},\tag{4.61}$$

where $\mathbf{w}_{\beta,k}$ (a subset of $\mathbf{w}_k$) is a small artificial noise term that allows the Kalman Filter to change its estimate of $\boldsymbol{\beta}_k$. The choice of $\mathbf{w}_{\beta,k}$ is arbitrarily chosen and it affects the convergence rate of the parameter estimation. A larger value allows faster convergence, but also larger oscillations around the real value, while a smaller value leads to slower convergence, but also smaller oscillations around the real value.

From an algorithmic perspective, the filter is initialized with an estimate of the mean value of the state $\mathbf{x}_0^+ = E(\mathbf{x}_0)$, and of its covariance matrix $\mathbf{P}_0^+ = E[(\mathbf{x}_0 - \mathbf{x}_0^+)(\mathbf{x}_0 -$

$\mathbf{x}_0^+)^T]$ (an initial guess). Then, the following operations are performed for $k = 1, 2, \dots$:

$$\begin{aligned} \mathbf{P}_k^- &= \mathbf{A}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{A}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{w,k-1} \mathbf{L}_{k-1}^T, \\ \mathbf{x}_k^- &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}), \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{S}_k^T (\mathbf{S}_k \mathbf{P}_k^- \mathbf{S}_k^T + \mathbf{M}_k \mathbf{Q}_{v,k} \mathbf{M}_k^T)^{-1}, \\ \mathbf{x}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k^-, \mathbf{0})), \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{S}_k) \mathbf{P}_k^-, \end{aligned} \quad (4.62)$$

where $\mathbf{A}_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}$, $\mathbf{L}_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{w}_k}$, $\mathbf{S}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}$ and $\mathbf{M}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{v}_k}$ are derived from the linearization of the process and observation models around the current state and noise.

To apply the model described by Eqs. (4.60), (4.61), and (4.62) to the initial-pose self-calibration problem, we consider a state update formulation (function $\mathbf{f}_k$) similar to the one defined in [128], in which acceleration and angular velocity measurements from an *IMU* mounted on the platform are used. While cable length variations and swivel angle variations can be used as output $\mathbf{y}_k$ for state correction. In particular, the augmented state $\mathbf{x}_k$, the input $\mathbf{u}_k$, the output vector $\mathbf{y}_k$, the process noise $\mathbf{w}_k$, and the measurement noise $\mathbf{v}_k$ can be set equal to:

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} \mathbf{p}_k \\ \dot{\mathbf{p}}_k \\ \boldsymbol{\epsilon}_k \\ \zeta_{0k} \end{bmatrix} \in \mathbb{R}^{15}, \quad \mathbf{u}_k = \begin{bmatrix} \ddot{\mathbf{p}}_{IMU,k} - \mathbf{w}_{1,k} \\ \boldsymbol{\omega}_{IMU,k} - \mathbf{w}_{2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{1,k} \\ \mathbf{u}_{2,k} \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} \Delta \mathbf{l}_k + \mathbf{v}_{l,k} \\ \Delta \boldsymbol{\sigma}_k + \mathbf{v}_{\sigma,k} \end{bmatrix}, \\ \mathbf{w}_k &= \begin{bmatrix} \mathbf{w}_{1,k} \\ \mathbf{w}_{2,k} \\ \mathbf{w}_{\zeta_{0,k}} \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} \mathbf{v}_{l,k} \\ \mathbf{v}_{\sigma,k} \end{bmatrix}, \end{aligned} \quad (4.63)$$

where $\ddot{\mathbf{p}}_{IMU}$ and $\boldsymbol{\omega}_{IMU}$ are the acceleration and angular velocity measured by the *IMU*, expressed in its local frame. While the state update $\mathbf{f}_{k-1}$ and the measurement update $\mathbf{h}_k$ are defined as:

$$\mathbf{f}_{k-1} = \begin{bmatrix} \mathbf{p}_{k-1} + \delta t \dot{\mathbf{p}}_{k-1} \\ \dot{\mathbf{p}}_{k-1} + \delta t (\mathbf{R}_{k-1} (\mathbf{u}_{1,k-1} + \mathbf{w}_{1,k-1}) - \mathbf{g}) \\ \boldsymbol{\epsilon}_{k-1} + \delta t \mathbf{H}_{k-1}^{-1} \mathbf{R}_{k-1} (\mathbf{u}_{2,k-1} + \mathbf{w}_{2,k-1}) \\ \zeta_{0,k-1} + \mathbf{w}_{\zeta_{0,k}} \end{bmatrix}, \quad (4.64)$$

$$\mathbf{h}_k = \begin{bmatrix} \mathbf{l}(\zeta_k) - \mathbf{l}_0(\zeta_0) + \mathbf{v}_{l,k} \\ \boldsymbol{\sigma}(\zeta_k) - \boldsymbol{\sigma}_0(\zeta_0) + \mathbf{v}_{\sigma,k} \end{bmatrix}, \quad (4.65)$$

with δt being the sampling time, and recalling Chapter 2, $\mathbf{R}_k = \mathbf{R}(\boldsymbol{\epsilon}_k)$ is the rotation matrix from the local mobile platform frame to the inertial frame, $\mathbf{H}_k = \mathbf{H}(\boldsymbol{\epsilon}_k)$ is the matrix that relates the time derivative of the Euler angles to the angular velocity, and $\mathbf{g}$ is the gravity vector.

Therefore, in this specific case, the linearization around $\mathbf{x}_k$, $\mathbf{w}_k$, and $\mathbf{v}_k$ leads to:

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \delta t \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & -\delta t (\mathbf{R}_k (\mathbf{u}_{1,k} + \mathbf{w}_{1,k}))^\sim \mathbf{H}_k & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} - \delta t \left(\mathbf{H}_k^{-1} \frac{\partial \mathbf{H}_k}{\partial \boldsymbol{\epsilon}_k} \mathbf{H}_k^{-1} \mathbf{R}_k (\mathbf{u}_{2,k} + \mathbf{w}_{2,k}) + \mathbf{H}_k^{-1} (\mathbf{R}_k (\mathbf{u}_{2,k} + \mathbf{w}_{2,k}))^\sim \mathbf{H}_k \right) & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{I}_{6 \times 6} \end{bmatrix}, \quad (4.66)$$

$$\mathbf{L}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 6} \\ \delta t \mathbf{R}_k & \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \delta t \mathbf{H}_k^{-1} \mathbf{R}_k & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{I}_{6 \times 6} \end{bmatrix}, \quad \mathbf{S}_k = \begin{bmatrix} \mathbf{J}_{l_p}(\zeta_k) & \mathbf{0}_{n \times 3} & \mathbf{J}_{l_r}(\zeta_k) & -\mathbf{J}_l(\zeta_0) \\ \mathbf{J}_{\sigma_p}(\zeta_k) & \mathbf{0}_{n \times 3} & \mathbf{J}_{\sigma_r}(\zeta_k) & -\mathbf{J}_\sigma(\zeta_0) \end{bmatrix}, \quad \mathbf{M}_k = \mathbf{I}_{2n \times 2n}, \quad (4.67)$$

where $\mathbf{J}_{l_p}$, $\mathbf{J}_{l_r}$, $\mathbf{J}_{\sigma_p}$, and $\mathbf{J}_{\sigma_r}$ are the positional and rotational blocks of matrices $\mathbf{J}_l$ and $\mathbf{J}_\sigma$ respectively, and $(\cdot)^\sim$ indicates here the skew-symmetric representation of the vector inside the parentheses.

The primary advantage of the presented *EKF* approach is the seamless integration of high-frequency *IMU* data with the system's nonlinear kinematics, providing a more robust estimate of the platform pose. However, because the *EKF* relies on first-order Taylor series expansions to propagate the covariance, the resulting $\mathbf{P}_k$ matrix may be less representative of the true uncertainty than in the unscented case, particularly in regions of high nonlinearity. Consequently, while the *PDOP* and *ODOP* indices are here derived from the diagonal of $\mathbf{P}_k^+$ for performance evaluation, they are subject to the inherent limitations of the linearization process. In terms of computational efficiency, the *EKF* is significantly less demanding than the unscented version, as it requires only a single Jacobian evaluation per iteration. This makes it a highly attractive candidate for real-time implementation, provided that the filter is properly tuned to maintain consistency between the estimated uncertainty and the actual error norms.

4.4.3.1 Unscented version

Given the n_x -state discrete-time nonlinear system defined in Eq. (4.60), the Unscented Kalman Filter (*UKF*) provides a method to estimate the state $\mathbf{x}_k$ without the need for linearization, by using the *UT* to propagate the mean and covariance of the state distribution through the nonlinear functions (4.64) and (4.65). The *UKF* algorithm can be summarized in the following steps [124, 129].

The filter is initialized with the state estimate and covariance:

$$\hat{\mathbf{x}}_0^+ = E(\mathbf{x}_0), \quad \mathbf{P}_0^+ = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]. \quad (4.68)$$

For each time step k , let $L = n_x + n_w + n_v$ be the dimension of the augmented state. The *UKF* algorithm proceeds as follows:

1. Augment the state and covariance to include process and measurement noise:

$$\check{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \\ \mathbf{v}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_k^+ \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \check{\mathbf{P}}_k^+ = \begin{bmatrix} \mathbf{P}_k^+ & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{w,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{v,k} \end{bmatrix}. \quad (4.69)$$

2. Generate augmented sigma points:

$$\begin{aligned} \check{\mathbf{x}}^{(i)} &= \check{\mathbf{x}}_{k-1}^+ + \bar{\mathbf{x}}^{(i)}, \quad i = 1, \dots, 2L, \\ \bar{\mathbf{x}}^{(i)} &= \left(\sqrt{n \check{\mathbf{P}}_{k-1}^+} \right)_i^T, \quad i = 1, \dots, L, \\ \bar{\mathbf{x}}^{(n+i)} &= - \left(\sqrt{n \check{\mathbf{P}}_{k-1}^+} \right)_i^T, \quad i = 1, \dots, L. \end{aligned} \quad (4.70)$$

3. Extract the state and noise components from the sigma points:

$$\mathbf{x}_{k-1}^{(i)} = \check{\mathbf{x}}_{1:n_x}^{(i)}, \quad \mathbf{w}_{k-1}^{(i)} = \check{\mathbf{x}}_{n_x+1:n_x+n_w}^{(i)}, \quad \mathbf{v}_k^{(i)} = \check{\mathbf{x}}_{n_x+n_w+1:L}^{(i)} \quad (4.71)$$

where n_w and n_v are the dimensions of the process and measurement noise, respectively.

4. Propagate the sigma points through the process model:

$$\mathbf{x}_k^{(i)-} = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}^{(i)}), \quad i = 1, \dots, 2L. \quad (4.72)$$

5. Compute the predicted state mean and covariance:

$$\hat{\mathbf{x}}_k^- = \frac{1}{2L} \sum_{i=1}^{2L} \mathbf{x}_k^{(i)-}, \quad \mathbf{P}_k^- = \frac{1}{2L} \sum_{i=1}^{2L} \left(\mathbf{x}_k^{(i)-} - \hat{\mathbf{x}}_k^- \right) \left(\mathbf{x}_k^{(i)-} - \hat{\mathbf{x}}_k^- \right)^T. \quad (4.73)$$

6. Propagate the sigma points through the measurement model:

$$\mathbf{y}_k^{(i)} = \mathbf{h}_k(\mathbf{x}_k^{(i)-}, \mathbf{v}_k^{(i)}), \quad i = 1, \dots, 2L. \quad (4.74)$$

7. Compute the predicted measurement mean and covariance:

$$\hat{\mathbf{y}}_k = \frac{1}{2L} \sum_{i=1}^{2L} \mathbf{y}_k^{(i)}, \quad \mathbf{P}_y = \frac{1}{2L} \sum_{i=1}^{2L} \left(\mathbf{y}_k^{(i)} - \hat{\mathbf{y}}_k \right) \left(\mathbf{y}_k^{(i)} - \hat{\mathbf{y}}_k \right)^T. \quad (4.75)$$

8. Compute the cross-covariance matrix:

$$\mathbf{P}_{xy} = \frac{1}{2L} \sum_{i=1}^{2L} \left(\mathbf{x}_k^{(i)-} - \hat{\mathbf{x}}_k^- \right) \left(\mathbf{y}_k^{(i)} - \hat{\mathbf{y}}_k \right)^T. \quad (4.76)$$

9. Compute the correction:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{xy} \mathbf{P}_y^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k) \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T. \end{aligned} \quad (4.77)$$

The *UKF* was applied to the same self-calibration problem defined in Sec. 4.4.3, using the same definitions for $\mathbf{x}_k$, $\mathbf{u}_k$, $\mathbf{y}_k$, $\mathbf{w}_k$, $\mathbf{v}_k$ (Eq. (4.63)), $\mathbf{f}_{k-1}$ and $\mathbf{h}_k$ (Eqs. (4.64) and (4.65)).

Similar to the *EKF*, the *UKF* provides real-time *PDOP* and *ODOP* indices derived from the diagonal of $\mathbf{P}_k^+$. While the *UKF* offers superior handling of the nonlinearities present in the *CDPR*'s geometry and IMU integration, it does so at a significantly higher computational cost, requiring $2L$ model evaluations per time step.

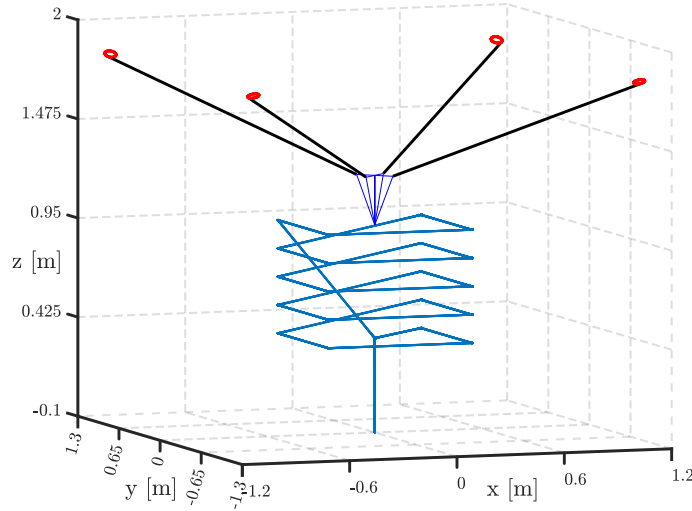


Figure 4.32: Trajectory executed for the online self-calibration data acquisition.

4.4.4 Results

A preliminary study was conducted by applying the aforementioned online methods to the *IRMA-4U UACDPR* introduced in Sec. 4.3.5. Ground truth data was collected using a Vicon motion capture system, following the same procedure described in Sec. 4.3.5.

No specific optimization procedure was employed for the measurement configurations or trajectory. Instead, an arbitrary trajectory was identified to cover the majority of the workspace, approaching the boundaries to enhance observability while maintaining safety margins, following the principles outlined in Sec. 4.2.2. The executed trajectory is shown in Fig. 4.32.

The sensor setup matches that described in Sec. 4.3.5, with the *AHRS* providing acceleration and rate-of-turn measurements (gyroscope accuracy: $0.2^\circ/\text{s}$; accelerometer accuracy: 0.007 g). Data was sampled at $f_s = 100\text{ Hz}$, resulting in a discrete time step of $\delta t = 0.01\text{ s}$ for the Kalman filter implementations. Computation was performed offline in MATLAB 2023b on a PC with an Intel i7-13650HX CPU and 32 GB of RAM.

The measurement-noise covariance matrix was assumed diagonal, $\mathbf{R}_k = \text{diag}(\sigma_l^2, \dots, \sigma_m^2)$, with parameters detailed in Table 4.11. For the *NESTED* method, the

$\sigma_l^2 [\text{m}^2]$	$\sigma_\sigma^2 [\text{deg}^2]$	$\sigma_\phi^2 [\text{deg}^2]$	$\sigma_\theta^2 [\text{deg}^2]$	$\sigma_\chi^2 [\text{deg}^2]$	$\sigma_{\dot{p}_{IMU}}^2 [(\text{m}/\text{s}^2)^2]$	$\sigma_{\omega_{IMU}}^2 [\text{rad}/\text{s}^2]$
0.004^2	0.2^2	0.2^2	0.2^2	0.4^2	0.0075^2	0.1^2

Table 4.11: Sensor noise parameters.

data buffer dimension was set to 30 measurements, with *PDOP* and *ODOP* thresholds of 10 mm and 2 deg, respectively. The tuning parameter α for the *RLS* methods was set to 0.75 for positional components and 28 for orientation. The initial *a posteriori* covariance and noise matrices for the *KF* algorithms were initialized as:

$$\mathbf{P}_0^+ = \text{diag}(0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.2, 0.2, 0.2, 0.5, 0.5, 0.5). \quad (4.78)$$

$$\begin{aligned} \mathbf{Q}_{w,k} &= \text{diag}(\sigma_{\dot{p}_{IMU}}^2, \dots, \sigma_{\dot{p}_{IMU}}^2, \sigma_{\omega_{IMU}}^2, \dots, \sigma_{\omega_{IMU}}^2, 1\text{E-}8, \dots, 1\text{E-}8), \\ \mathbf{Q}_{v,k} &= \text{diag}(\sigma_l^2, \sigma_l^2, \sigma_l^2, \sigma_\sigma^2, \sigma_\sigma^2, \sigma_\sigma^2). \end{aligned} \quad (4.79)$$

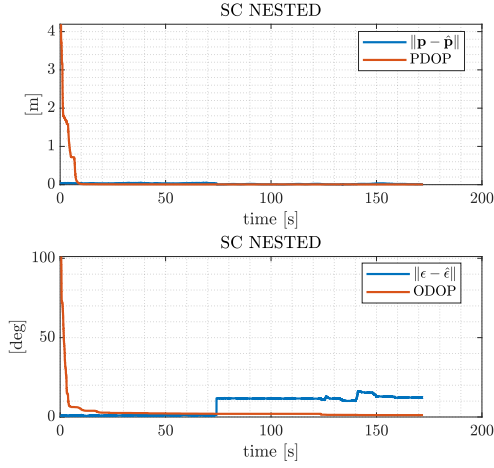


Figure 4.33: *PDOP* and *ODOP* nested method.

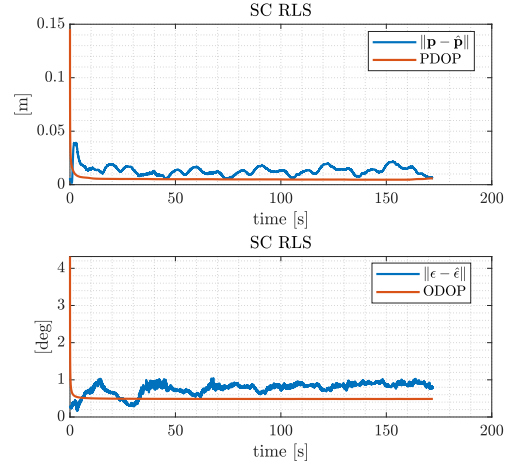


Figure 4.34: *PDOP* and *ODOP* *RLS* method.

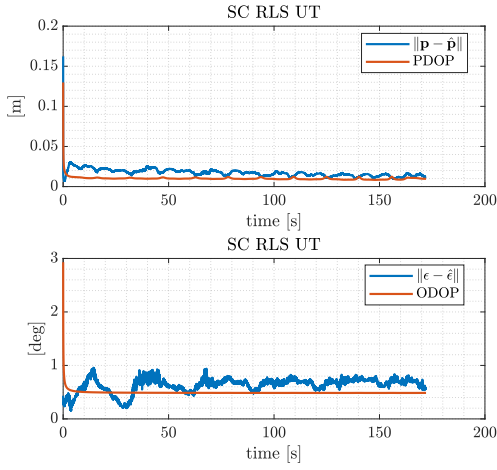


Figure 4.35: *PDOP* and *ODOP* *RLS* unscented method.

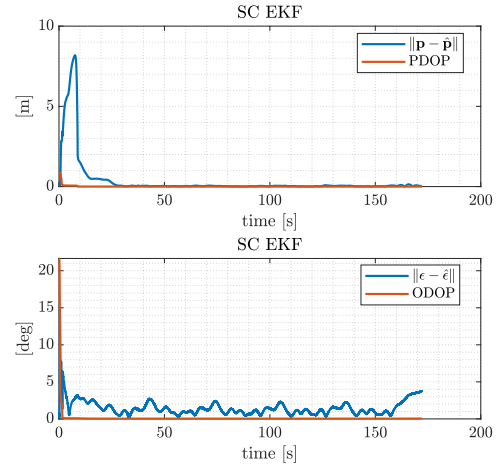


Figure 4.36: *PDOP* and *ODOP* *EKF* method.

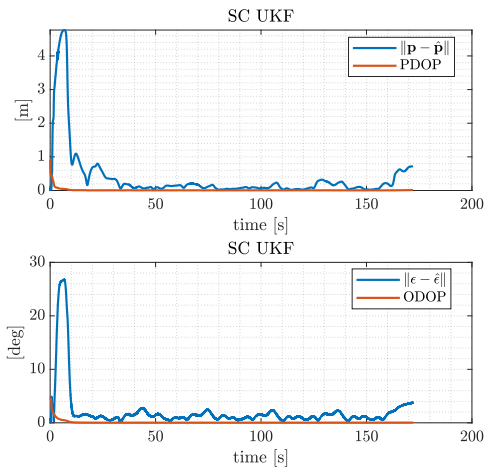


Figure 4.37: *PDOP* and *ODOP* *EKF* method.

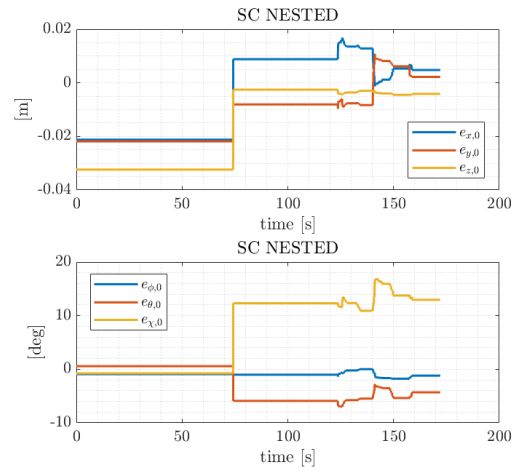
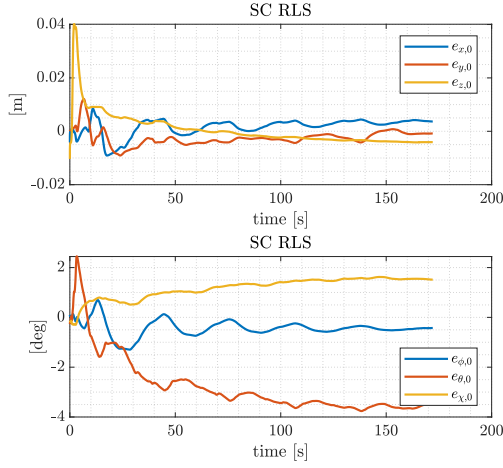
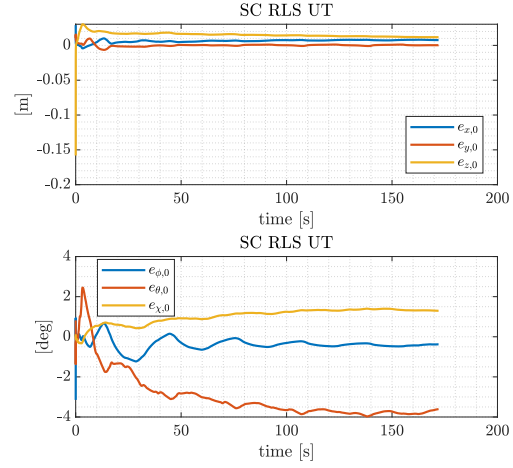
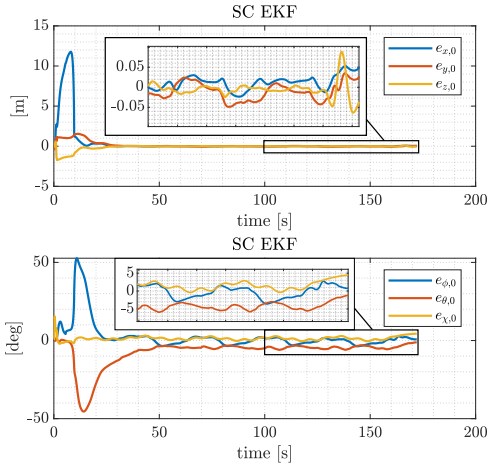
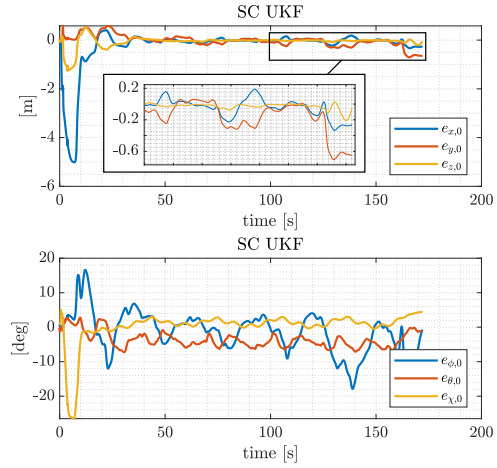


Figure 4.38: Initial pose error components, nested method.


 Figure 4.39: Initial pose error components, *RLS* method.

 Figure 4.40: Initial pose error components, *RLS* unscented method.

 Figure 4.41: Initial pose error components, *EKF* method.

 Figure 4.42: Initial pose error components, *UKF* method.

The relationship between observability indices and actual error is illustrated in Figs. 4.33 to 4.37. In the *NESTED* method (Fig. 4.33), *PDOP* and *ODOP* drop sharply within the first 10 seconds, suggesting rapid convergence. However, the *ODOP* fails to anticipate the significant jump in orientation error $\|\epsilon - \hat{\epsilon}\|$ occurring at approximately 75 seconds, highlighting a limitation where the index becomes "blind" to drift after the initial calibration window.

Conversely, the unscented versions—both *RLS UT* (Fig. 4.35) and *UKF* (Fig. 4.37), demonstrate superior stochastic consistency. The *UKF* effectively captures the massive initial transient spike (reaching 5 m in position and 25 deg in orientation) before settling after 25 seconds. While the *RLS* methods show very low and flat observability indices, they surprisingly track the modest oscillations in the position error norm ($\|\mathbf{p} - \hat{\mathbf{p}}\| < 0.05$ m) quite well despite the partial covariance updates.

In Figs. 4.38 to 4.42 the error in the initial pose estimate is shown for every component over time. The evolution of the initial position error components ($e_{x,0}$, $e_{y,0}$, $e_{z,0}$) across all tested methods shows a general trend toward zero after an initial transient period. The *RLS*-based approaches exhibit the smoothest parameter evolution, with

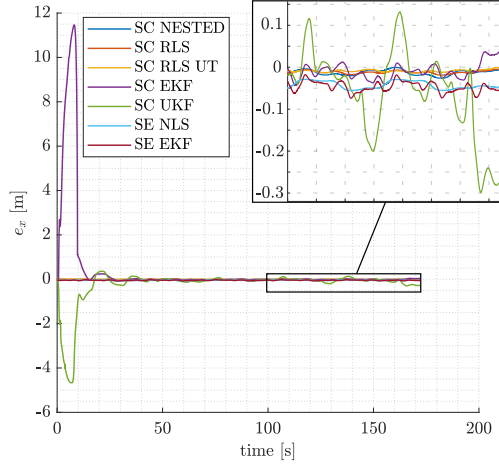


Figure 4.43: Comparison of the online x error.

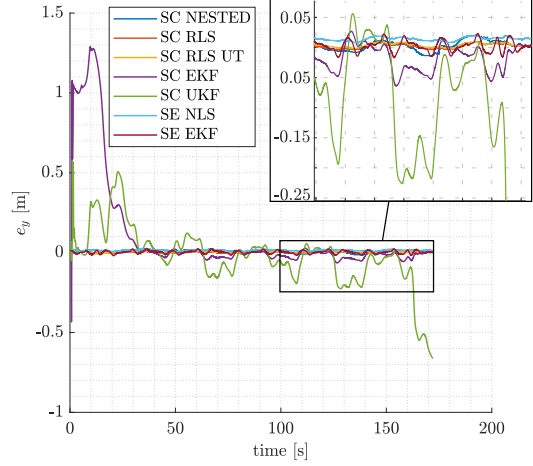


Figure 4.44: Comparison of the online y error.

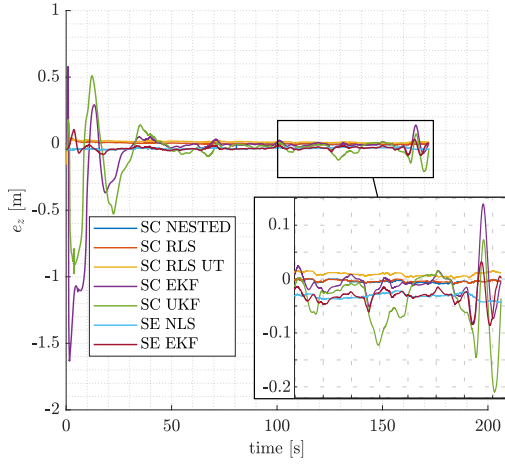


Figure 4.45: Comparison of the online z error.

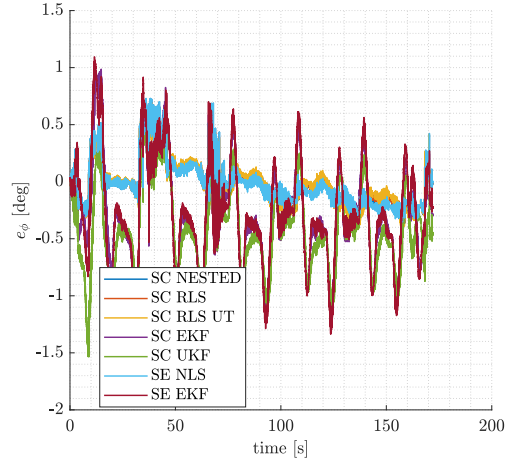


Figure 4.46: Comparison of the online roll error.

the *RLS UT* variant demonstrating noticeably faster convergence—settling within approximately 15 seconds—compared to the 50 seconds required for the standard *RLS*. In contrast, the *NESTED* method updates its parameters in discrete steps at approximately 75, 125, and 140 seconds, as the data buffer is processed and the outer optimization loop is executed. The *EKF* and *UKF* implementations are characterized by high initial transients; for instance, $e_{x,0}$ spikes to nearly 12 m in the *EKF* and -5 m in the *UKF* as the filters reconcile the high-frequency *IMU* inputs with the kinematic model before stabilizing after 50 to 100 seconds. The steady state error of the *UKF* remains high compared to the linearized version.

Orientation calibration proved significantly more challenging, as the *NESTED* and *RLS* approaches were unable to maintain long-term stability across all Euler angle components. The *NESTED* method shows the poorest performance in this regard, with the yaw error ($e_{\chi,0}$) jumping to approximately 15° at the 75-second mark and remaining uncorrected for the rest of the trajectory. The *EKF* and *UKF* experience massive initial swings in Euler angles (with roll and yaw errors exceeding 25° during the transient phase), and they keep non-negligible errors at steady state.

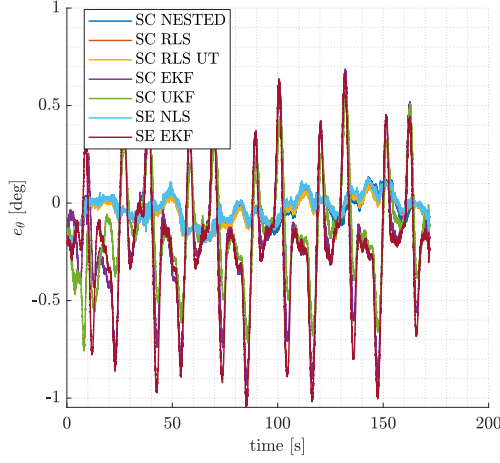


Figure 4.47: Comparison of the online pitch error.

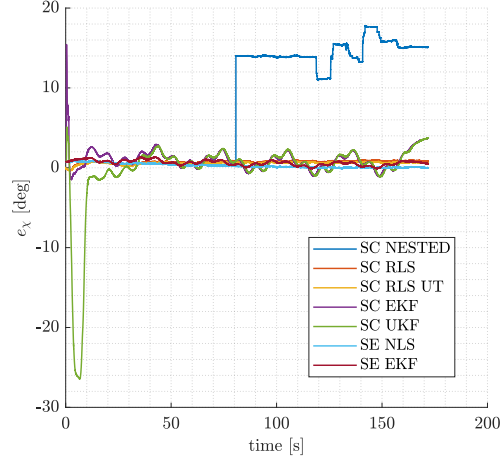


Figure 4.48: Comparison of the online yaw error.

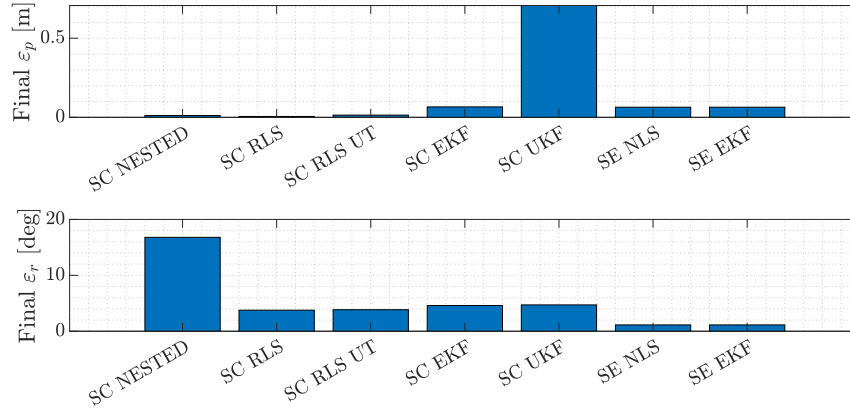


Figure 4.49: Comparison of the last initial pose estimation errors.

Even though this work aims at obtaining a good estimate of the initial pose, we also report the results of the current pose estimation errors, as it is part of the self-calibration solution. Therefore, the graphs from Fig. 4.43 to Fig. 4.48 show the instantaneous error in the EE pose estimate. To properly assess the actual impact of the presented online self-calibration methods, these charts include state estimation executed without ever performing the initial pose self-calibration (i.e., the initial pose estimate is kept unchanged). In detail, the state estimation is performed with the method of Sec. 4.4.1 without the outer loop ($SE\ NLS$) and with the EKF (Sec. 4.4.3) without augmenting the state vector with the initial pose ($SE\ EKF$).

The instantaneous results highlight the drawback of online parameter estimation. The $SC\ EKF$ and $SC\ UKF$ show very high initial positional errors (peaking at 8 m and -5 m in e_x , respectively) as the filters struggle to resolve the initial pose offsets. Once settled, they provide estimates comparable to the baselines. However, the $SC\ UKF$ demonstrates a concerning late-trajectory drift in the y component, which is reflected in the final error metrics. In the orientation domain, the $SC\ NESTED$ yaw error (e_χ) mirrors the failure of its initial pose calibration, maintaining a sustained offset of $\approx 12^\circ$ that significantly degrades the overall pose reconstruction.

The final performance of the various methods regarding the initial pose calibration

is summarized in the bar chart of Fig. 4.49. This chart compares the final positional error (ϵ_p) and rotational error (ϵ_r) for the calibration parameters ζ_0 at the end of the trajectory. Most methods, including the *SC NESTED*, *SC RLS*, and *SC RLS UT*, successfully drive the initial positional error to near-zero residuals. The *SC EKF* maintains a small residual, but the *SC UKF* exhibits a significant anomalous final error of approximately 0.7 m. This matches the late-trajectory divergence observed in its time-series data. The rotational results highlight the extreme sensitivity of the orientation parameters. Since the initial guess pose is very close to the actual initial pose, the *SE NLS* and *SE EKF* possess the lowest final rotational errors, approximately 1° . The online calibration methods generally increased this error: *SC RLS*, *SC RLS UT*, *SC EKF*, and *SC UKF* all concluded with rotational errors between 4° and 5° . The *SC NESTED* approach demonstrates the worst rotational performance, reaching a final error of nearly 17° . This confirms that the step-wise update based on an unoptimized trajectory can lead the calibration into a significant yaw offset that it cannot recover from.

Based on the comprehensive experimental results for the *IRMA-4U* prototype detailed throughout this chapter, a final qualitative comparison can be drawn between the manual, automatic, offline, and online techniques. The automatic technique incorporating tension measurements into the residuals (Sec. 4.2.2.4) provides the highest accuracy for both position and orientation estimates. However, it is important to note that these results may be susceptible to bias; single trials may not represent worst-case scenarios, and using internal state estimation as a reference may underestimate modeling errors—such as swivel-pulley friction—that external ground truth systems like Vicon are able to detect. The same condition affects the results of the manual calibration approach (Sec. 4.2.1.3), which, at first glance, seems to be a reliable baseline, particularly for orientation estimation. Online methods, specifically the *RLS* variants, excel in rapid positional convergence, outperforming offline methods in initial position recovery. Nevertheless, they generally exhibit lower performance in orientation estimation. The selection of a self-calibration method involves a direct trade-off between the desired accuracy and operational constraints: automatic offline methods are ideal for high-precision closed-environment settings where time allows for extensive data processing, manual methods provide a robust orientation baseline for initial deployment, online *RLS* or *EKF* are best suited for rapid, autonomous calibration in unconstrained environments, provided that the increased sensitivity to orientation drift is acceptable for the task.

Chapter 5

Conclusions

Cable-Driven Parallel Robots (*CDPRs*) utilize lightweight cables instead of rigid links to manipulate objects in space. While this architecture offers significant advantages in terms of workspace scalability and payload capacity, it introduces inherent uncertainties, such as cable sagging, elasticity, and friction, that limit the manipulator's operational precision. This thesis has focused on improving the accuracy and reliability of *CDPRs* by addressing both workspace analysis and system initialization. Specifically, a control-based workspace analysis was performed to identify high-performance regions where the *CDPR* can operate safely and accurately. Furthermore, as relative sensors are frequently employed for state estimation, the initial End-Effector (*EE*) pose is often highly uncertain at every machine startup. If unaddressed, this uncertainty propagates through the system, leading to significant errors in subsequent *EE* state estimation. Therefore, this research has analyzed and developed advanced self-calibration techniques for accurate initial-pose estimation.

5.1 Main results

Generic physical models for both classical and deployable *CDPR* architectures were first established, incorporating the geometry of swiveling pulley transmissions. These models provided the foundation for zero- to second-order kinematics and statics, which were essential for the subsequent workspace analysis and self-calibration algorithms.

A primary contribution of this work is the methodology developed for the performance analysis of Overconstrained *CDPRs* (*OCDPRs*). It was demonstrated that traditional sensitivity indices, often reduced to matrix norms, are insufficient to capture the combined impact of cable-length and tension-control errors when a Hybrid Control in Joint Space (*HC*) is adopted. To address this, the concept of the Wrench-Feasible Error-Insensitive Workspace (*WFEIW*) was introduced. This novel workspace subset ensures that user-specified positioning and orientation precision are maintained, and that cable tensions remain within predetermined safe bounds, despite realistic sensor noise and control perturbations. As a practical application, an analysis was conducted on various robot architectures, both classic and deployable, evaluating different sets of length- and force-controlled cables. Furthermore, an algorithm for platform design optimization and optimal selection of the combination of length- and force-controlled cables was presented. Such optimization may yield a significant increase in the *WFEIW* volume for a given installation footprint.

Regarding system initialization, novel data acquisition frameworks were introduced for the initial-pose self-calibration of Underactuated *CDPRs* (*UACDPRs*). Both manual and automatic procedures were proposed to support rapid data collection while maintaining high-quality initial-pose estimates. These approaches leverage pure force joint control to ensure cables remain tensioned, thereby keeping the geometric constraints of the system active. The automated procedure enables safe motion transitions between static equilibrium poses within the static workspace. By incorporating multiple sensor measurements—including cable tensions—into the calibration process, the number of required measurement poses is reduced without sacrificing estimation accuracy. Experimental validation confirmed that utilizing static equilibrium poses in the calibration formulation is superior for achieving accurate results. Specifically, the analysis demonstrated that leveraging force measurements effectively minimizes the number of required configurations, thus reducing total calibration time.

A comprehensive and systematic framework was developed for analyzing the influence of the main design parameters and algorithmic choices on the performance of initial-pose self-calibration procedures for *CDPRs*. By combining a Monte Carlo-based simulation methodology with a structured evaluation of sensor configurations, measurement pose selection, and nonlinear solvers, the proposed approach enables a quantitative assessment of accuracy, robustness, and computational efficiency prior to experimental deployment. The analysis, conducted on both an overconstrained and an underactuated *CDPR* architecture, highlighted that the sensor set represents the dominant factor affecting self-calibration performance, followed by the number of measurement poses, while the choice of the nonlinear solver mainly influences convergence speed and computational cost rather than final accuracy. In particular, increasing the number of measurement poses improves estimation accuracy up to a saturation point, and the inclusion of informative measurements, such as swivel-angle encoders and orientation sensing, is essential to achieve reliable results, especially for underactuated systems. Among the analyzed solvers, the Levenberg–Marquardt method consistently provides the most favorable compromise between robustness and computational efficiency, although all methods converge to comparable solutions when sufficient information is available. Experimental validation on two real robotic prototypes confirmed that the proposed analysis framework effectively predicts the main performance trends observed in practice, while also revealing the impact of unmodeled systematic effects and architectural differences on achievable accuracy. These results demonstrate that the proposed methodology constitutes a valuable tool for guiding the design and implementation of initial-pose self-calibration procedures in real *CDPR* systems.

Finally, novel online initial-pose self-calibration techniques were developed based on Recursive Least Squares and Kalman filtering. These methods represent a paradigm shift from traditional sequential offline approaches. The findings demonstrate that the online algorithms achieve a level of accuracy comparable to offline batch methods that utilize a large number of poses, but they do so in a significantly shorter timeframe. By performing calibration concurrently with the robot's initial movements, the machine setup time is minimized, enabling immediate operational readiness and facilitating deployment in unstructured or rapidly changing environments.

5.2 Open Issues

While effective approaches for motion control and trajectory planning during startup were established, managing the system when the *EE* pose is unknown remains a significant challenge for *OCDPRs*, with further room for improvement in *UACDPRs*. For *UACDPRs*, trajectory planning based on the solution of a modified Inverse Geometrico-Static Problem (*IGSP*) could be developed to optimize measurement configuration selection based on observability criteria.

The data acquisition process can be further improved by developing a unified tool that integrates automatic pose selection, motion planning, and control strategies to minimize both path length and transition time between equilibrium poses. Additionally, mixed self-calibration algorithms that fuse static or dynamic equations with geometrical constraints could be investigated. A key challenge in this integration involves designing algorithms for the automatic extraction of data from logged measurements, ensuring that static equations are applied only when the *EE* is in true equilibrium, while using the remaining sampled data for kinematic or dynamic formulations. In the context of online techniques, cable tensions could be further exploited through dynamic models and Kalman filtering, provided that a reliable estimate of inertial parameters is available.

Furthermore, the investigation of hybrid model-based and data-driven methods may offer new solutions to the self-calibration problem, particularly in accounting for non-ideal cable behaviors. Finally, the findings of this thesis can be extended into a general framework for the self-calibration and identification of all geometric and inertial parameters. Such a framework is essential for achieving the high-precision motion and operational performance required for the next generation of industrial *CDPRs*.

Appendices

Appendix A

Proofs

A.1 Computation of the swivel angle

Substituting Eq. (2.3) into the pulley coplanarity constraint (2.8) yields:

$$[-\sin(\sigma_i)\mathbf{i}_i + \cos(\sigma_i)\mathbf{j}_i] \cdot \boldsymbol{\rho}_i = 0. \quad (\text{A.1})$$

Rearranging the terms results in:

$$\sin(\sigma_i)(\mathbf{i}_i \cdot \boldsymbol{\rho}_i) = \cos(\sigma_i)(\mathbf{j}_i \cdot \boldsymbol{\rho}_i), \quad (\text{A.2})$$

which leads to Eq. (2.9):

$$\sigma_i = \text{atan2}(\mathbf{j}_i \cdot \boldsymbol{\rho}_i, \mathbf{i}_i \cdot \boldsymbol{\rho}_i) \quad (\text{A.3})$$

The use of the $\text{atan2}(\cdot)$ function is preferred as it is defined for all values of its arguments and ensures the swivel angle is uniquely determined within the interval $[-\pi, \pi]$

A.2 Computation of the tangency angle

If we consider:

$$\boldsymbol{\rho}_i = \mathbf{p} + \mathbf{a}_i - \mathbf{b}_i = \mathbf{p} + \mathbf{a}_i - (\mathbf{d}_i + r_i(\mathbf{u}_i + \mathbf{n}_i)) = \boldsymbol{\rho}_i - r_i(\mathbf{u}_i + \mathbf{n}_i), \quad (\text{A.4})$$

and substitute it in Eq. (2.11), we obtain:

$$\mathbf{n}_i \cdot (\boldsymbol{\rho}_i - r_i(\mathbf{u}_i + \mathbf{n}_i)) = \mathbf{n}_i \cdot \boldsymbol{\rho}_i - r_i(\mathbf{n}_i \cdot \mathbf{u}_i + 1) = 0. \quad (\text{A.5})$$

Then, substituting Eq. (2.4) in Eq. (A.5) and rearranging yields:

$$\begin{aligned} (\cos(\psi_i)\mathbf{u}_i + \sin(\psi_i)\mathbf{k}_i) \cdot \boldsymbol{\rho}_i - r_i((\cos(\psi_i)\mathbf{u}_i + \sin(\psi_i)\mathbf{k}_i) \cdot \mathbf{u}_i + 1) &= \\ = (\mathbf{u}_i \cdot \boldsymbol{\rho}_i - r_i)\cos(\psi_i) + \mathbf{k}_i \cdot \boldsymbol{\rho}_i \sin(\psi_i) - r_i &= \\ = (\rho_{u_i} - r_i)\cos(\psi_i) + \rho_{k_i} \sin(\psi_i) - r_i &= 0 \end{aligned} \quad (\text{A.6})$$

where $\rho_{k_i} = \mathbf{k}_i \cdot \boldsymbol{\rho}_i$ and $\rho_{u_i} = \mathbf{u}_i \cdot \boldsymbol{\rho}_i$. If the trigonometric identity:

$$\cos(\psi_i) = \frac{1 - t_i^2}{1 + t_i^2}, \quad \sin(\psi_i) = \frac{2t_i}{1 + t_i^2}, \quad t_i = \tan(\psi_i/2) \quad (\text{A.7})$$

is applied to Eq. (A.6), we get:

$$(\varrho_{u_i} - r_i)(1 - t_i^2) + 2\varrho_{k_i} t_i - r_i(1 + t_i^2) = -\varrho_{u_i} t_i^2 + 2\varrho_{k_i} t_i + \varrho_{u_i} - 2r_i = 0 \quad (\text{A.8})$$

which can be straightforwardly solved for t_i as:

$$t_i = \tan(\psi/2) = \frac{\varrho_{k_i}}{\varrho_{u_i}} \pm \sqrt{\left(\frac{\varrho_{k_i}}{\varrho_{u_i}}\right)^2 + 1 - \frac{2r_i}{\varrho_{u_i}}} \quad (\text{A.9})$$

Solving for t_i gives two possible solutions. As specified in Eq. (2.12), we select the positive root, which corresponds to the physical case of the cable being wrapped clockwise around the pulley groove:

$$\psi_i = 2 \operatorname{atan}\left(\frac{\varrho_{k_i}}{\varrho_{u_i}} + \sqrt{\left(\frac{\varrho_{k_i}}{\varrho_{u_i}}\right)^2 + 1 - \frac{2r_i}{\varrho_{u_i}}}\right) \quad (\text{A.10})$$

A.3 Computation of the swivel angle pose derivative

The time derivative of Eq. (2.8) yields:

$$\dot{\mathbf{w}}_i \cdot \boldsymbol{\varrho}_i + \mathbf{w}_i \cdot \dot{\boldsymbol{\varrho}}_i = 0. \quad (\text{A.11})$$

Since $\dot{\boldsymbol{\varrho}}_i = \dot{\mathbf{p}} + \dot{\mathbf{a}}_i$, we may compute the velocity of point D_i and the rate of change of $\mathbf{w}_i$ from Eqs. (2.7) and (2.3) as:

$$\dot{\boldsymbol{\varrho}}_i = \dot{\mathbf{p}} + \boldsymbol{\omega} \times \mathbf{a}_i, \quad \dot{\mathbf{w}}_i = -\dot{\sigma}_i \mathbf{u}_i. \quad (\text{A.12})$$

Substituting Eq. (A.12) in Eq. (A.11) and rearranging yields:

$$\mathbf{u}_i \cdot \boldsymbol{\varrho}_i \dot{\sigma}_i = \varrho_{u_i} \dot{\sigma}_i = \mathbf{w}_i \cdot \dot{\mathbf{p}} + \mathbf{w}_i \cdot \boldsymbol{\omega} \times \mathbf{a}_i = \dot{\mathbf{p}} \cdot \mathbf{w}_i + \boldsymbol{\omega} \cdot \mathbf{a}_i \times \mathbf{w}_i, \quad (\text{A.13})$$

thus:

$$\dot{\sigma}_i = \frac{1}{\varrho_{u_i}} (\dot{\mathbf{p}} \cdot \mathbf{w}_i + \boldsymbol{\omega} \cdot \mathbf{a}_i \times \mathbf{w}_i) = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{v}. \quad (\text{A.14})$$

Substituting Eq. (2.16) in Eq. (A.14) we obtain the relation between the i -th swivel angle time derivative and the pose time derivative:

$$\dot{\sigma}_i = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{D} \dot{\boldsymbol{\zeta}}. \quad (\text{A.15})$$

Therefore, using the chain rule, we obtain Eq. (2.23) from Eq. (A.15) as:

$$\frac{\partial \sigma_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{D}. \quad (\text{A.16})$$

A.4 Computation of the tangency angle pose derivative

The time derivative of $\mathbf{n}_i \cdot \boldsymbol{\rho}_i = 0$ yields:

$$\dot{\mathbf{n}}_i \cdot \boldsymbol{\rho}_i + \mathbf{n}_i \cdot \dot{\boldsymbol{\rho}}_i = 0. \quad (\text{A.17})$$

Its terms can be computed by differentiating w.r.t. time Eqs. (2.10), (2.2) and (2.4):

$$\dot{\boldsymbol{\rho}}_i = \dot{\mathbf{p}} + \dot{\mathbf{a}}_i - r_i (\dot{\mathbf{u}}_i + \dot{\mathbf{n}}_i), \quad \dot{\mathbf{u}}_i = \dot{\sigma}_i \mathbf{w}_i, \quad \dot{\mathbf{n}}_i = \cos(\psi_i) \dot{\sigma}_i \mathbf{w}_i - \dot{\psi}_i \mathbf{t}_i. \quad (\text{A.18})$$

Substituting Eq. (A.18) in Eq. (A.17) and rearranging yields:

$$\mathbf{t}_i \cdot \boldsymbol{\rho}_i \dot{\psi}_i = \|\boldsymbol{\rho}_i\| \dot{\psi}_i = \mathbf{n}_i \cdot \dot{\mathbf{p}} + \mathbf{n}_i \cdot \boldsymbol{\omega} \times \mathbf{a}_i = \dot{\mathbf{p}} \cdot \mathbf{n}_i + \boldsymbol{\omega} \cdot \mathbf{a}_i \times \mathbf{n}_i. \quad (\text{A.19})$$

since $\mathbf{w}_i \cdot \mathbf{n}_i = \mathbf{t}_i \cdot \mathbf{n}_i = \boldsymbol{\rho}_i \cdot \mathbf{n}_i = 0$.

Thus:

$$\dot{\psi}_i = \frac{1}{\|\boldsymbol{\rho}_i\|} (\dot{\mathbf{p}} \cdot \mathbf{n}_i + \boldsymbol{\omega} \cdot \mathbf{a}_i \times \mathbf{n}_i) = \boldsymbol{\xi}_{\psi_i} \cdot \mathbf{v}. \quad (\text{A.20})$$

with $\boldsymbol{\xi}_{\psi_i} = \frac{1}{\|\boldsymbol{\rho}_i\|} \begin{bmatrix} \mathbf{n}_i \\ \mathbf{a}_i \times \mathbf{n}_i \end{bmatrix}$.

Substituting Eq. (2.16) in Eq. (A.20) we obtain the relation between the i -th tangency angle time derivative and the pose time derivative:

$$\dot{\psi}_i = \boldsymbol{\xi}_{\psi_i}^T \mathbf{D} \dot{\boldsymbol{\zeta}}. \quad (\text{A.21})$$

A.5 Computation of the cable length pose derivative

The time derivative of the identity $\boldsymbol{\rho}_i^T \boldsymbol{\rho}_i = \|\boldsymbol{\rho}_i\|^2$ yields:

$$2\dot{\boldsymbol{\rho}}_i \cdot \boldsymbol{\rho}_i - 2\|\boldsymbol{\rho}_i\| \frac{\partial \|\boldsymbol{\rho}_i\|}{\partial t} = 0. \quad (\text{A.22})$$

The time derivative of $\|\boldsymbol{\rho}_i\|$ is computed from Eq. (2.13) as:

$$\frac{\partial \|\boldsymbol{\rho}_i\|}{\partial t} = \dot{l}_i + r_i \dot{\psi}_i. \quad (\text{A.23})$$

Substituting Eqs. (A.18) and (A.23) in Eq. (A.22), and considering that $\boldsymbol{\rho}_i = \|\boldsymbol{\rho}_i\| \mathbf{t}_i$ yields:

$$(\dot{\mathbf{p}} + \dot{\mathbf{a}}_i) \cdot \|\boldsymbol{\rho}_i\| \mathbf{t}_i - r_i \|\boldsymbol{\rho}_i\| [(\cos(\psi_i) + 1) \dot{\sigma}_i \mathbf{w}_i - \dot{\psi}_i \mathbf{t}_i] \cdot \mathbf{t}_i - \|\boldsymbol{\rho}_i\| (\dot{l}_i + r_i \dot{\psi}_i) = 0. \quad (\text{A.24})$$

Simplifying and rearranging Eq. (A.24) gives:

$$\dot{l}_i = (\dot{\mathbf{p}} \cdot \mathbf{t}_i + \boldsymbol{\omega} \cdot \mathbf{a}_i \times \mathbf{t}_i) = \boldsymbol{\xi}_{l_i} \cdot \mathbf{v}. \quad (\text{A.25})$$

Substituting Eq. (2.16) in Eq. (A.25) we obtain the relation between the i -th cable length time derivative and the pose time derivative:

$$\dot{l}_i = \boldsymbol{\xi}_{l_i}^T \mathbf{D} \dot{\boldsymbol{\zeta}}. \quad (\text{A.26})$$

Therefore, using the chain rule, we obtain Eq. (2.24) from Eq. (A.26) as:

$$\frac{\partial l_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{l_i}^T \mathbf{D}. \quad (\text{A.27})$$

A.6 Computation of the swivel angle second-order pose derivative

The time derivative of Eq. (A.13) yields:

$$(\dot{\mathbf{u}}_i \cdot \boldsymbol{\rho}_i + \mathbf{u}_i \cdot \dot{\boldsymbol{\rho}}_i) \dot{\sigma}_i + \rho_{u_i} \ddot{\sigma}_i = \dot{\mathbf{w}}_i \cdot (\dot{\mathbf{p}} + \dot{\mathbf{a}}_i) + \mathbf{w}_i \cdot (\ddot{\mathbf{p}} + \boldsymbol{\alpha} \times \mathbf{a}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i)), \quad (\text{A.28})$$

and accounting for Eqs. (A.12), (A.18) one has:

$$(\mathbf{u}_i \cdot (\dot{\mathbf{p}} + \boldsymbol{\omega} \times \mathbf{a}_i)) \dot{\sigma}_i + \rho_{u_i} \ddot{\sigma}_i = \dot{\sigma}_i \mathbf{u}_i \cdot (\dot{\mathbf{p}} + \boldsymbol{\omega} \times \mathbf{a}_i) + \mathbf{w}_i \cdot (\ddot{\mathbf{p}} + \boldsymbol{\alpha} \times \mathbf{a}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i)), \quad (\text{A.29})$$

where the term $\dot{\sigma}_i^2 \mathbf{w}_i \boldsymbol{\rho}_i$ was omitted due to Eq. (2.8).

Isolating $\ddot{\sigma}_i$, and substituting $\dot{\sigma}_i$ with Eq. (A.14), we obtain:

$$\ddot{\sigma}_i = -\frac{1}{\rho_{u_i}} (-\mathbf{v}^T \boldsymbol{\xi}_{u_i} \boldsymbol{\xi}_{\sigma_i}^T \mathbf{v} - \mathbf{v}^T \boldsymbol{\xi}_{\sigma_i} \boldsymbol{\xi}_{u_i}^T \mathbf{v} + \mathbf{w}_i \cdot \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i)) + \boldsymbol{\xi}_{\sigma_i}^T \dot{\mathbf{v}}, \quad \text{with} \quad \boldsymbol{\xi}_{u_i} = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{a}_i \times \mathbf{u}_i \end{bmatrix}. \quad (\text{A.30})$$

Thanks to the vector triple-product identity, one has:

$$\mathbf{w}_i \cdot \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i) = \boldsymbol{\omega} \cdot \mathbf{a}_i \times (\mathbf{w}_i \times \boldsymbol{\omega}), \quad (\text{A.31})$$

and Eq. (A.30) can be rewritten in matrix form as:

$$\ddot{\sigma}_i = \mathbf{v}^T \bar{\boldsymbol{\xi}}_{\sigma_i} \mathbf{v} + \boldsymbol{\xi}_{\sigma_i}^T \dot{\mathbf{v}}, \quad (\text{A.32})$$

where:

$$\bar{\boldsymbol{\xi}}_{\sigma_i} = \frac{1}{\rho_{u_i}} (-\boldsymbol{\xi}_{u_i} \boldsymbol{\xi}_{\sigma_i}^T - \boldsymbol{\xi}_{\sigma_i} \boldsymbol{\xi}_{u_i}^T + \mathbf{A}_{w_i}), \quad \mathbf{A}_{w_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{a}}_i \tilde{\mathbf{w}}_i \end{bmatrix}. \quad (\text{A.33})$$

It should be noted that:

$$\ddot{\sigma}_i = \dot{\boldsymbol{\xi}}_{\sigma_i}^T \mathbf{v} + \boldsymbol{\xi}_{\sigma_i} \cdot \dot{\mathbf{v}}, \quad \text{and} \quad \dot{\boldsymbol{\xi}}_{\sigma_i}^T = \mathbf{v}^T \bar{\boldsymbol{\xi}}_{\sigma_i} = \dot{\boldsymbol{\zeta}}^T \mathbf{D}^T \bar{\boldsymbol{\xi}}_{\sigma_i}. \quad (\text{A.34})$$

From Eq. (A.34), using the chain rule, we finally obtain the derivative of $\boldsymbol{\xi}_{\sigma_i}^T$ with respect to the pose:

$$\frac{\partial \boldsymbol{\xi}_{\sigma_i}^T}{\partial \boldsymbol{\zeta}} = \mathbf{D}^T \bar{\boldsymbol{\xi}}_{\sigma_i}. \quad (\text{A.35})$$

A.7 Computation of the cable length second-order pose derivative

The time derivative of Eq. (A.25) yields:

$$\ddot{l}_i = \dot{\mathbf{t}}_i \cdot (\dot{\mathbf{p}} + \dot{\mathbf{a}}_i) + \mathbf{t}_i \cdot (\ddot{\mathbf{p}} + \boldsymbol{\alpha} \times \mathbf{a}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i)), \quad (\text{A.36})$$

where $\dot{\mathbf{t}}_i$ can be evaluated as the time derivative of Eq. (2.5) as:

$$\dot{\mathbf{t}}_i = \sin(\psi_i) \dot{\sigma}_i \mathbf{w}_i + \dot{\psi}_i \mathbf{n}_i. \quad (\text{A.37})$$

Then, according to Eqs. (A.37) and (A.12), Eq. (A.36) becomes:

$$\ddot{l}_i = (\sin \psi_i \dot{\sigma}_i \mathbf{w}_i + \dot{\psi}_i \mathbf{n}_i) \cdot (\dot{\mathbf{p}} + \boldsymbol{\omega} \times \mathbf{a}_i) + \mathbf{t}_i \cdot (\ddot{\mathbf{p}} + \boldsymbol{\alpha} \times \mathbf{a}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i)). \quad (\text{A.38})$$

Then, substituting $\dot{\sigma}_i$ and $\dot{\psi}_i$ with Eqs. (A.14) and (A.20), respectively, we obtain:

$$\ddot{l}_i = \sin(\psi_i) \rho_{u_i} \mathbf{v}^T \xi_{\sigma_i} \xi_{\sigma_i}^T \mathbf{v} + \|\rho_i\| \mathbf{v}^T \xi_{\psi_i} \xi_{\psi_i}^T \mathbf{v} + \mathbf{t}_i \cdot \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i) + \xi_{l_i} \cdot \dot{\mathbf{v}}, \quad (\text{A.39})$$

which can be written in matrix form as:

$$\ddot{l}_i = \mathbf{v}^T \xi'_{l_i} \mathbf{v} + \xi_{l_i}^T \dot{\mathbf{v}}. \quad (\text{A.40})$$

where:

$$\bar{\xi}_{l_i} = \sin(\psi_i) \rho_{u_i} \xi_{\sigma_i} \xi_{\sigma_i}^T + \|\rho_i\| \xi_{\psi_i} \xi_{\psi_i}^T + \mathbf{A}_{t_i}, \quad \mathbf{A}_{t_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \tilde{\mathbf{a}}_i \tilde{\mathbf{t}}_i \end{bmatrix}. \quad (\text{A.41})$$

It should be noted that:

$$\ddot{l}_i = \dot{\xi}_{l_i}^T \mathbf{v} + \xi_{l_i}^T \dot{\mathbf{v}}, \quad \dot{\xi}_{l_i}^T = \mathbf{v}^T \bar{\xi}_{l_i} = \dot{\zeta}^T \mathbf{D}^T \bar{\xi}_{l_i}. \quad (\text{A.42})$$

From Eq. (A.42), using the chain rule, we finally obtain the derivative of $\xi_{l_i}^T$ with respect to the pose:

$$\frac{\partial \xi_{l_i}^T}{\partial \zeta} = \mathbf{D}^T \bar{\xi}_{l_i}. \quad (\text{A.43})$$

A.8 Computation of the swivel angle pose derivative for deployable CDPRs

The time derivative of Eq. (2.43) yields:

$$\dot{\mathbf{w}}_i \cdot \boldsymbol{\rho}_i + \mathbf{w}_i \cdot \dot{\boldsymbol{\rho}}_i = 0, \quad (\text{A.44})$$

in which the derivative of $\mathbf{w}_i$ is obtained deriving Eq. (2.39) (considering Eq. (2.35)):

$$\dot{\mathbf{w}}_i = -\dot{\sigma}_i \mathbf{u}_i + \boldsymbol{\omega} \times \mathbf{w}_i. \quad (\text{A.45})$$

While the derivative of $\boldsymbol{\rho}_i$ is computed from Eq. (2.42) as

$$\dot{\boldsymbol{\rho}}_i = -\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i. \quad (\text{A.46})$$

Substituting Eqs. (A.45) and (A.46) in (A.44) we obtain

$$-\dot{\sigma}_i \mathbf{u}_i \cdot \boldsymbol{\rho}_i + (\boldsymbol{\omega} \times \mathbf{w}_i) \cdot \boldsymbol{\rho}_i + \mathbf{w}_i \cdot (-\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i) = 0. \quad (\text{A.47})$$

Isolating $\dot{\sigma}$, we have:

$$\dot{\sigma} = \frac{1}{\mathbf{u}_i \cdot \boldsymbol{\rho}_i} \left((\boldsymbol{\omega} \times \mathbf{w}_i) \cdot \boldsymbol{\rho}_i + \mathbf{w}_i \cdot (-\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i) \right). \quad (\text{A.48})$$

Simplifying and rearranging yields

$$\dot{\sigma} = \frac{1}{\mathbf{u}_i \cdot \boldsymbol{\rho}_i} \left(-\mathbf{w}_i \cdot \dot{\mathbf{p}} - ((\boldsymbol{\rho}_i + \mathbf{d}_i) \times \mathbf{w}_i) \cdot \boldsymbol{\omega} \right) = \xi_{\sigma_i} \mathbf{v}. \quad (\text{A.49})$$

Substituting Eq. (2.16) in Eq. (A.49) we obtain the relation between the i -th swivel angle time derivative and the pose time derivative:

$$\dot{\sigma}_i = \xi_{\sigma_i}^T \mathbf{D} \dot{\zeta}. \quad (\text{A.50})$$

Therefore, using the chain rule, we obtain Eq. (2.50) from Eq. (A.50) as:

$$\frac{\partial \sigma_i}{\partial \zeta} = \xi_{\sigma_i}^T \mathbf{D}. \quad (\text{A.51})$$

A.9 Computation of the tangency angle pose derivative for deployable CDPRs

The time derivative of Eq. (2.46) yields:

$$\dot{\mathbf{n}}_i \cdot \boldsymbol{\rho}_i + \mathbf{n}_i \cdot \dot{\boldsymbol{\rho}}_i = 0. \quad (\text{A.52})$$

Its terms can be computed by differentiating w.r.t. time Eqs. (2.40), (2.36), (2.38), (2.34), and (2.45):

$$\dot{\boldsymbol{\rho}}_i = -\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i - r_i(\dot{\mathbf{u}}_i + \dot{\mathbf{n}}_i), \quad \dot{\mathbf{u}}_i = \dot{\sigma}_i \mathbf{w}_i + \boldsymbol{\omega} \times \mathbf{u}_i, \quad \dot{\mathbf{n}}_i = \cos(\psi_i) \dot{\sigma}_i \mathbf{w}_i - \dot{\psi}_i \mathbf{t}_i + \boldsymbol{\omega} \times \mathbf{n}_i. \quad (\text{A.53})$$

Substituting Eq. (A.53) in Eq. (A.52) results in:

$$\begin{aligned} & (\cos(\psi_i) \dot{\sigma}_i \mathbf{w}_i - \dot{\psi}_i \mathbf{t}_i + \boldsymbol{\omega} \times \mathbf{n}_i) \cdot \boldsymbol{\rho}_i + \\ & \mathbf{n}_i \cdot (-\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i - r_i(\dot{\sigma}_i \mathbf{w}_i + \boldsymbol{\omega} \times \mathbf{u}_i + \cos(\psi_i) \dot{\sigma}_i \mathbf{w}_i - \dot{\psi}_i \mathbf{t}_i + \boldsymbol{\omega} \times \mathbf{n}_i)) = 0. \end{aligned} \quad (\text{A.54})$$

Considering that $\boldsymbol{\rho}_i = \mathbf{t}_i \|\boldsymbol{\rho}_i\|$, simplifying and rearranging yields:

$$-\dot{\psi}_i \|\boldsymbol{\rho}_i\| + \|\boldsymbol{\rho}_i\| \mathbf{w}_i \cdot \boldsymbol{\omega} - \mathbf{n}_i \cdot \dot{\mathbf{p}} - (\mathbf{d}_i \times \mathbf{n}_i) \cdot \boldsymbol{\omega} - r_i(\mathbf{u}_i \times \mathbf{n}_i) \cdot \boldsymbol{\omega} = 0 \quad (\text{A.55})$$

since $\mathbf{w}_i \cdot \mathbf{n}_i = \mathbf{t}_i \cdot \mathbf{n}_i = \boldsymbol{\rho}_i \cdot \mathbf{n}_i = 0$.

Isolating $\dot{\psi}_i$ gives:

$$\dot{\psi}_i = \frac{1}{\|\boldsymbol{\rho}_i\|} (-\mathbf{n}_i \cdot \dot{\mathbf{p}} - ((\mathbf{d}_i + r_i \mathbf{u}_i) \times \mathbf{n}_i + \|\boldsymbol{\rho}_i\| \mathbf{w}_i) \cdot \boldsymbol{\omega}) = \boldsymbol{\xi}_{\psi_i} \cdot \mathbf{v}, \quad (\text{A.56})$$

with

$$\boldsymbol{\xi}_{\psi_i} = \frac{1}{\|\boldsymbol{\rho}_i\|} \begin{bmatrix} -\mathbf{n}_i \\ -(\mathbf{d}_i + r_i \mathbf{u}_i) \times \mathbf{n}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{w}_i \end{bmatrix}. \quad (\text{A.57})$$

Substituting Eq. (2.16) in Eq. (A.56) we obtain the relation between the i -th tangency angle time derivative and the pose time derivative:

$$\dot{\psi}_i = \boldsymbol{\xi}_{\psi_i}^T \mathbf{D} \dot{\boldsymbol{\zeta}}. \quad (\text{A.58})$$

A.10 Computation of the cable length pose derivative for deployable CDPRs

The time derivative of the identity $\boldsymbol{\rho}_i^T \boldsymbol{\rho}_i = \|\boldsymbol{\rho}_i\|^2$ yields:

$$2\dot{\boldsymbol{\rho}}_i \cdot \boldsymbol{\rho}_i - 2\|\boldsymbol{\rho}_i\| \frac{\partial \|\boldsymbol{\rho}_i\|}{\partial t} = 0. \quad (\text{A.59})$$

The time derivative of $\|\boldsymbol{\rho}_i\|$ is computed from Eq. (2.48) as:

$$\frac{\partial \|\boldsymbol{\rho}_i\|}{\partial t} = \dot{l}_i + r_i \dot{\psi}_i. \quad (\text{A.60})$$

Substituting Eqs. (A.53) and (A.60) in Eq. (A.59) we obtain:

$$(-\dot{\mathbf{p}} - \boldsymbol{\omega} \times \mathbf{d}_i - r_i(\dot{\sigma}_i \mathbf{w}_i(1 + \cos \psi_i) + \boldsymbol{\omega} \times (\mathbf{u}_i + \mathbf{n}_i) - \dot{\psi}_i \mathbf{t}_i)) \cdot \boldsymbol{\rho}_i - \|\boldsymbol{\rho}_i\|(\dot{l}_i + r_i \dot{\psi}_i) = 0. \quad (\text{A.61})$$

Simplifying and rearranging Eq.(A.61) gives:

$$\dot{l}_i = -\mathbf{t}_i \cdot \dot{\mathbf{p}} - ((\mathbf{d}_i + r_i(\mathbf{u}_i + \mathbf{n}_i)) \times \mathbf{t}_i) \cdot \boldsymbol{\omega} = \boldsymbol{\xi}_{l_i}^T \mathbf{v}. \quad (\text{A.62})$$

Substituting Eq. (2.16) in Eq. (A.62) we obtain the relation between the i -th cable length time derivative and the pose time derivative:

$$\dot{l}_i = \boldsymbol{\xi}_{l_i}^T \mathbf{D} \dot{\boldsymbol{\zeta}}. \quad (\text{A.63})$$

Therefore, using the chain rule, we obtain Eq. (2.51) from Eq. (A.63) as:

$$\frac{\partial l_i}{\partial \boldsymbol{\zeta}} = \boldsymbol{\xi}_{l_i}^T \mathbf{D}. \quad (\text{A.64})$$

A.11 Computation of the cable length second-order pose derivative for deployable CDPRs

To derive the Hessian of the cable length for the deployable architecture, we define the auxiliary vector $\mathbf{q}_i$ as the relative position from the platform origin P to the pulley exit point B_i . From Eq. (2.51), the relationship between the cable length rate and the twist $\mathbf{v}$ becomes:

$$\dot{l}_i = \begin{bmatrix} -\mathbf{t}_i \\ -\mathbf{q}_i \times \mathbf{t}_i \end{bmatrix}^T \mathbf{v}. \quad (\text{A.65})$$

Differentiating Eq. (A.65) with respect to time yields:

$$\ddot{l}_i = \begin{bmatrix} -\dot{\mathbf{t}}_i \\ -\dot{\mathbf{q}}_i \times \mathbf{t}_i - \mathbf{q}_i \times \dot{\mathbf{t}}_i \end{bmatrix}^T \mathbf{v} + \begin{bmatrix} -\mathbf{t}_i \\ -\mathbf{q}_i \times \mathbf{t}_i \end{bmatrix}^T \dot{\mathbf{v}} = \boldsymbol{\xi}_{l_i}^T \dot{\mathbf{v}} + \dot{\boldsymbol{\xi}}_{l_i}^T \mathbf{v}, \quad (\text{A.66})$$

where the time derivatives of the unit vector $\mathbf{t}_i$ and the exit point vector $\mathbf{q}_i$ are:

$$\dot{\mathbf{t}}_i = \dot{\psi}_i \mathbf{n}_i + \dot{\sigma}_i \mathbf{w}_i \sin \psi_i + \boldsymbol{\omega} \times \mathbf{t}_i \quad (\text{A.67})$$

$$\dot{\mathbf{q}}_i = \boldsymbol{\omega} \times \mathbf{q}_i + r_i (\dot{\sigma}_i \mathbf{w}_i (1 + \cos \psi_i) - \dot{\psi}_i \mathbf{t}_i). \quad (\text{A.68})$$

By substituting Eqs. (A.67) and (A.68) into the time derivative of $\boldsymbol{\xi}_{l_i}$, we obtain:

$$\dot{\boldsymbol{\xi}}_{l_i} = \begin{bmatrix} -\dot{\psi}_i \mathbf{n}_i - \dot{\sigma}_i \mathbf{w}_i \sin \psi_i - \boldsymbol{\omega} \times \mathbf{t}_i \\ -(\boldsymbol{\omega} \times \mathbf{q}_i + r_i \dot{\sigma}_i \mathbf{w}_i (1 + \cos \psi_i)) \times \mathbf{t}_i - \mathbf{q}_i \times (\dot{\psi}_i \mathbf{n}_i + \dot{\sigma}_i \mathbf{w}_i \sin \psi_i + \boldsymbol{\omega} \times \mathbf{t}_i) \end{bmatrix}. \quad (\text{A.69})$$

Grouping common terms and substituting joint rates with their geometric Jacobian expressions ($\dot{\psi}_i = \boldsymbol{\xi}_{\psi_i}^T \mathbf{v}$ and $\dot{\sigma}_i = \boldsymbol{\xi}_{\sigma_i}^T \mathbf{v}$), the expression becomes:

$$\dot{\boldsymbol{\xi}}_{l_i} = \dot{\psi}_i \begin{bmatrix} -\mathbf{n}_i \\ \mathbf{q}_i \times \mathbf{n}_i \end{bmatrix} + \sin \psi_i \dot{\sigma}_i \begin{bmatrix} -\mathbf{w}_i \\ -\mathbf{q}_i \times \mathbf{w}_i \end{bmatrix} + (r_i + r_i \cos \psi_i) \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{t}_i \times \mathbf{w}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} & \tilde{\mathbf{t}}_i \\ \mathbf{0}_{3 \times 3} & -\tilde{\mathbf{t}}_i \tilde{\mathbf{q}}_i + \tilde{\mathbf{q}}_i \tilde{\mathbf{t}}_i \end{bmatrix} \mathbf{v}, \quad (\text{A.70})$$

thus, finally

$$\dot{\boldsymbol{\xi}}_{l_i} = \boldsymbol{\xi}_{n_i} \boldsymbol{\xi}_{\psi_i}^T \mathbf{v} + \sin \psi_i \boldsymbol{\xi}_{w_i} \boldsymbol{\xi}_{\sigma_i}^T \mathbf{v} + (r_i + r_i \cos \psi_i) \boldsymbol{\xi}_{t_i} \boldsymbol{\xi}_{\sigma_i}^T \mathbf{v} + \mathbf{L}_i \mathbf{v} = \bar{\boldsymbol{\xi}}_{l_i}^T \mathbf{v} \quad (\text{A.71})$$

where

$$\begin{aligned} \boldsymbol{\xi}_{n_i} &= \begin{bmatrix} -\mathbf{n}_i \\ \mathbf{q}_i \times \mathbf{n}_i \end{bmatrix}, \quad \boldsymbol{\xi}_{w_i} = \begin{bmatrix} -\mathbf{w}_i \\ -\mathbf{q}_i \times \mathbf{w}_i \end{bmatrix}, \quad \boldsymbol{\xi}_{t_i} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{t}_i \times \mathbf{w}_i \end{bmatrix}, \quad \mathbf{L}_i = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \tilde{\mathbf{t}}_i \\ \mathbf{0}_{3 \times 3} & -\tilde{\mathbf{t}}_i \tilde{\mathbf{q}}_i + \tilde{\mathbf{q}}_i \tilde{\mathbf{t}}_i \end{bmatrix}, \\ \bar{\boldsymbol{\xi}}_{l_i}^T &= \boldsymbol{\xi}_{n_i} \boldsymbol{\xi}_{\psi_i}^T + \sin \psi_i \boldsymbol{\xi}_{w_i} \boldsymbol{\xi}_{\sigma_i}^T + (r_i + r_i \cos \psi_i) \boldsymbol{\xi}_{t_i} \boldsymbol{\xi}_{\sigma_i}^T + \mathbf{L}_i. \end{aligned} \quad (\text{A.72})$$

Since

$$\dot{\boldsymbol{\xi}}_{l_i}^T = \mathbf{v}^T \bar{\boldsymbol{\xi}}_{l_i} = \dot{\boldsymbol{\zeta}}^T \mathbf{D}^T \bar{\boldsymbol{\xi}}_{l_i}, \quad (\text{A.73})$$

from Eq. (A.73), using the chain rule, we finally obtain the derivative of $\boldsymbol{\xi}_{l_i}^T$ with respect to the pose:

$$\frac{\partial \boldsymbol{\xi}_{l_i}^T}{\partial \boldsymbol{\zeta}} = \mathbf{D}^T \bar{\boldsymbol{\xi}}_{l_i}. \quad (\text{A.74})$$

List of Author Publications

1. E. Idà, F. Zoffoli, and M. Carricato. (2024). Hybrid-Control-Based Workspace Analysis of Overconstrained Cable-Driven Parallel Robots. In: Lenarčič, J., Husty, M. eds. *Advances in Robot Kinematics 2024 (ARK 2024)*, pp. 324-331, Springer, 2024, doi: 10.1007/978-3-031-64057-5_37.
2. F. Zoffoli, V. Coccia, E. Idà, M. Carricato, A rapid initial-pose self-calibration method for underactuated cable-driven parallel robots, In: G. Quaglia, G. Boschetti, G. Carbone (eds), *Advances in Italian Mechanism Science, IFToMM Italy 2024, Mechanisms and Machine Science*, pp. 366-374, 2024, doi:10.1007/978-3-031-64553-2_43.
3. F. Zoffoli, E. Idà, M. Carricato, Design and control optimization for hybrid-controlled overconstrained cable-driven parallel robots, *Mechanism and Machine Theory*, 209, pp. 1-17, 2024, doi:10.1016/j.mechmachtheory.2025.105998.
4. F. Zoffoli, E. Idà, M. Carricato, Initial-pose self-calibration for deployable overconstrained Cable-Driven Parallel Robots, In: K. Jovanović, A. Rodić, M. Raković (eds) *Advances in Service and Industrial Robotics, RAAD 2025, Mechanisms and Machine Science*, vol 190. Springer, Cham, pp. 559-567, 2025, doi:10.1007/978-3-032-02106-9_62
5. F. Zoffoli, E. Idà, M. Carricato, A comparative analysis of initial-pose self-calibration algorithms for underactuated cable-driven parallel robot, *Robotica*, Cambridge University Press, pp. 1-17, 2025, doi:10.1017/S0263574725102804

List of Figures

1.1	Serial (left) and parallel (right) robot architecture: the <i>EE</i> is connected to the base via an open chain (serial) or via multiple closed chains (parallel) composed of links and joints.	13
1.2	Examples of robot manipulators: (a) serial and (b) parallel.	13
1.3	Classical robot applications.	14
1.4	Annual installations of industrial robots by customer industry.	14
1.5	Robotic applications in building construction.	15
1.6	Model of a CDPR: the <i>moving platform</i> is connected to the <i>fixed frame</i> by <i>cables</i> , which are uncoiled by a set of <i>servo-winch</i> es and guided through <i>pulley systems</i>	16
1.7	CDPRs applications	17
1.8	CDPRs applications in building construction and maintenance.	18
2.1	Geometric model of the i -th cable transmission.	24
2.2	i -th swivel pulley geometric model	25
2.3	<i>EE</i> Free-Body Diagram	28
2.4	Geometric model of the i -th cable transmission for deployable CDPRs.	29
2.5	i -th swivel pulley geometric model	30
2.6	<i>EE</i> Free-Body Diagram for deployable CDPRs	32
3.1	IPAnema 3 OCDPR	44
3.2	HRP OCDPR	44
3.3	IRMA-8 deployable OCDPR	45
3.4	IPAnema 3 Workspace with 1-2 as force-controlled cables	47
3.5	IPAnema 3 Workspace with 4-6 as force-controlled cables	47
3.6	HRP Workspace with 2-8 as force-controlled cables	48
3.7	HRP Workspace with 2-7 as force-controlled cables	48
3.8	IRMA-8 Workspace with 1-2 as force-controlled cables	48
3.9	IRMA-8 Workspace with 2-3 as force-controlled cables	49
3.10	HRP Workspace with 2-8 as force-controlled cables. Cross sections in x-z planes at $y = -1.04\text{m}$, $y = -0.40\text{m}$, $y = 0.25\text{m}$, $y = 0.90\text{m}$	49
3.11	Nested optimization algorithm	51
3.12	Optimized geometry of the IPAnema 3 platform with $\delta\tau = 1\%\tau_{max}$	53
3.13	IPAnema 3 platform design optimization with $\delta\tau = 1\%\tau_{max}$	53
3.14	IPAnema 3 WFEIW before optimization, with $\delta\tau = 5\%\tau_{max}$	53
3.15	IPAnema 3 platform design optimization with $\delta\tau = 5\%\tau_{max}$	53
3.16	Optimized geometry of the IPAnema 3 platform with $\delta\tau = 5\%\tau_{max}$	54
4.1	Calibration poses selection and control planning	65

4.2	Flowchart of the data-acquisition algorithm	66
4.3	Experimental setup	68
4.4	Sampled and reference tensions with $k = 8$. The black curves, with superscript ($\circ$), represent the tension reference applied during transitions, while the colored curves are the measured tensions, with superscript ($*$).	70
4.5	Schematic of the prototype	74
4.6	Calibrated initial pose errors	74
4.7	Position error ε_p simulation results for the <i>OCDPR</i> prototype.	84
4.8	Position error ε_p simulation results for the <i>UACDPR</i> prototype.	84
4.9	Orientation error ε_r simulation results for the <i>OCDPR</i> prototype.	85
4.10	Orientation error ε_r simulation results for the <i>UACDPR</i> prototype.	85
4.11	Number of iterations n_{iter} simulation results for the <i>OCDPR</i> prototype.	86
4.12	Number of iterations n_{iter} simulation results for the <i>UACDPR</i> prototype.	86
4.13	Computation time t_{comp} simulation results for the <i>OCDPR</i> prototype.	87
4.14	Computation time t_{comp} simulation results for the <i>UACDPR</i> prototype.	87
4.15	Final cost function value f_{sol} simulation results for the <i>OCDPR</i> prototype.	88
4.16	Final cost function value f_{sol} simulation results for the <i>UACDPR</i> prototype.	88
4.17	Inlier rate simulation results for the <i>OCDPR</i> prototype.	89
4.18	Inlier rate simulation results for the <i>UACDPR</i> prototype.	89
4.19	Position error ε_p experiments results for the <i>OCDPR</i> prototype.	90
4.20	Orientation error ε_r experiments results for the <i>OCDPR</i> prototype.	90
4.21	Number of iterations n_{iter} experiments results for the <i>OCDPR</i> prototype.	91
4.22	Computation time t_{comp} experiments results for the <i>OCDPR</i> prototype.	91
4.23	Final cost function value f_{sol} experiments results for the <i>OCDPR</i> prototype.	91
4.24	Inlier rate experiments results for the <i>OCDPR</i> prototype.	91
4.25	Position error ε_p experiments results for the <i>UACDPR</i> prototype.	92
4.26	Orientation error ε_r experiments results for the <i>UACDPR</i> prototype.	92
4.27	Number of iterations n_{iter} experiments results for the <i>UACDPR</i> prototype.	92
4.28	Computation time t_{comp} experiments results for the <i>UACDPR</i> prototype.	93
4.29	Final cost function value f_{sol} experiments results for the <i>UACDPR</i> prototype.	93
4.30	Inlier rate experiments results for the <i>UACDPR</i> prototype.	93
4.31	Flowchart of the nested online initial-pose self-calibration algorithm.	96
4.32	Trajectory executed for the online self-calibration data acquisition.	103
4.33	<i>PDOP</i> and <i>ODOP</i> nested method.	104
4.34	<i>PDOP</i> and <i>ODOP</i> <i>RLS</i> method.	104
4.35	<i>PDOP</i> and <i>ODOP</i> <i>RLS</i> unscented method.	104
4.36	<i>PDOP</i> and <i>ODOP</i> <i>EKF</i> method.	104
4.37	<i>PDOP</i> and <i>ODOP</i> <i>EKF</i> method.	104
4.38	Initial pose error components, nested method.	104
4.39	Initial pose error components, <i>RLS</i> method.	105
4.40	Initial pose error components, <i>RLS</i> unscented method.	105
4.41	Initial pose error components, <i>EKF</i> method.	105
4.42	Initial pose error components, <i>UKF</i> method.	105
4.43	Comparison of the online x error.	106
4.44	Comparison of the online y error.	106
4.45	Comparison of the online z error.	106
4.46	Comparison of the online roll error.	106

4.47 Comparison of the online pitch error.	107
4.48 Comparison of the online yaw error.	107
4.49 Comparison of the last initial pose estimation errors.	107

Bibliography

- [1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer London, 2010.
- [2] B. Siciliano and O. Khatib, eds., *Springer Handbook of Robotics*. Springer Handbooks, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [3] B. Siciliano, L. Villani, G. Oriolo, and A. De Luca, *Foundations of Robotics*. Springer Nature Switzerland, 2025.
- [4] W. Khalil and E. Dombre, *Modeling identification & control of robots*. London [u.a.]: Kogan Page Science, new ed., kogan page science paper edition, digitaler nachdr ed., 2006. Includes bibliographical references (p. [447]-473) and index.
- [5] J. Merlet, *Parallel Robots*. Solid Mechanics and Its Applications, Springer Netherlands, 2012.
- [6] M. Gharbia, A. Chang-Richards, Y. Lu, R. Y. Zhong, and H. Li, “Robotic technologies for on-site building construction: A systematic review,” *Journal of Building Engineering*, vol. 32, p. 101584, Nov. 2020.
- [7] C. Gosselin, R. Duballet, P. Roux, N. Gaudillière, J. Dirrenberger, and P. Morel, “Large-scale 3d printing of ultra-high performance concrete – a new processing route for architects and builders,” *Materials & Design*, vol. 100, pp. 102–109, June 2016.
- [8] M. Trovinger, L. Giacobbe, T. Grantham, C. Sun, and J. Chen, “Reinforcement learning with vision-based state estimation for generalized robotic assembly,” in *Proceedings of the 42nd International Symposium on Automation and Robotics in Construction*, ISARC2025, International Association for Automation and Robotics in Construction (IAARC), July 2025.
- [9] Y. Wu, H. H. Cheng, A. Fingrut, K. Crolla, Y. Yam, and D. Lau, “Cu-brick cable-driven robot for automated construction of complex brick structures: From simulation to hardware realisation,” in *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, pp. 166–173, IEEE, May 2018.
- [10] E. Asadi, B. Li, and I.-M. Chen, “Pictobot: A cooperative painting robot for interior finishing of industrial developments,” *IEEE Robotics & Automation Magazine*, vol. 25, pp. 82–94, June 2018.

- [11] A. N. F. Chan, W. Cheng, and D. Lau, "Deformable open-frame cable-driven parallel robots: Modeling, analysis, and control," *IEEE Transactions on Robotics*, vol. 40, pp. 3465–3480, 2024.
- [12] A. Lucarini and E. Idá, *Kinematic Modeling and Design of a Sensorized Cable-Routing System for Cable-Driven Parallel Robots*, pp. 77–85. Springer International Publishing, 2022.
- [13] L. Zhang, J. Sun, P. Jiang, and H. Li, "Structural optimization and performance analysis of the cable-driven fine adjustment mechanism for the new fast feed cabin," *Scientific Reports*, vol. 16, Dec. 2025.
- [14] H. Tan, R. Muntashir, L. Nurahmi, B. Pramujati, A. Kurniawan, U. Wasiwitono, and S. Caro, "Stability analysis of a reconfigurable and mobile cable-driven parallel robot," *IEEE Access*, vol. 12, pp. 14182–14193, 2024.
- [15] J. Albus, R. Bostelman, and N. Dagalakis, "The nist robocrane," *Journal of Robotics System*, 1992-09-08 00:09:00 1992.
- [16] Z. Zhang, Z. Shao, Z. You, X. Tang, B. Zi, G. Yang, C. Gosselin, and S. Caro, "State-of-the-art on theories and applications of cable-driven parallel robots," *Frontiers of Mechanical Engineering*, vol. 17, Sept. 2022.
- [17] H. Hussein, J. C. Santos, and M. Gouttefarde, "Geometric optimization of a large scale cdpr operating on a building facade," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5117–5124, 2018.
- [18] K. Iturralde, M. Feucht, D. Illner, R. Hu, W. Pan, T. Linner, T. Bock, I. Eskudero, M. Rodriguez, J. Gorrotxategi, J. Izard, J. Astudillo, J. Cavalcanti Santos, M. Gouttefarde, M. Fabritius, C. Martin, T. Henninge, S. Nornes, Y. Jacobsen, A. Pracucci, J. Cañada, J. Jimenez-Vicaria, R. Alonso, and L. Elia, "Cable-driven parallel robot for curtain wall module installation," *Automation in Construction*, vol. 138, p. 104235, 2022.
- [19] E. Moreira, A. M. Pinto, P. Costa, A. P. Moreira, G. Veiga, J. Lima, J. P. Sousa, and P. Costa, "Cable robot for non-standard architecture and construction: A dynamic positioning system," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 3184–3189, IEEE, Mar. 2015.
- [20] T. Bruckmann, C. Reichert, M. Meik, P. Lemmen, A. Spengler, H. Mattern, and M. König, *Concept Studies of Automated Construction Using Cable-Driven Parallel Robots*, pp. 364–375. Springer International Publishing, July 2017.
- [21] T. Bruckmann, A. J. Spengler, C. K. Karl, C. Reichert, and M. König, *Process Analysis of Cable-Driven Parallel Robots for Automated Construction*, pp. 63–83. Springer International Publishing, 2018.
- [22] T. Bruckmann and R. Boumann, "Simulation and optimization of automated masonry construction using cable robots," *Advanced Engineering Informatics*, vol. 50, p. 101388, Oct. 2021.

-
- [23] P. Bosscher, R. L. Williams, L. S. Bryson, and D. Castro-Lacouture, "Cable-suspended robotic contour crafting system," *Automation in Construction*, vol. 17, pp. 45–55, Nov. 2007.
- [24] J.-B. Izard, A. Dubor, P.-E. Hervé, E. Cabay, D. Culla, M. Rodriguez, and M. Barado, *On the Improvements of a Cable-Driven Parallel Robot for Achieving Additive Manufacturing for Construction*, pp. 353–363. Springer International Publishing, July 2017.
- [25] P. C. Chesser, P. L. Wang, J. E. Vaughan, R. F. Lind, and B. K. Post, "Kinematics of a cable-driven robotic platform for large-scale additive manufacturing," *Journal of Mechanisms and Robotics*, vol. 14, Sept. 2021.
- [26] Z. Shao, G. Xie, Z. Zhang, and L. Wang, "Design and analysis of the cable-driven parallel robot for cleaning exterior wall of buildings," *International Journal of Advanced Robotic Systems*, vol. 18, Jan. 2021.
- [27] J.-P. Merlet and D. Daney, "A portable, modular parallel wire crane for rescue operations," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2834–2839, IEEE, May 2010.
- [28] Z. Zake and S. Caro, "Elasto-static modelling and identification of a deployable cable-driven parallel robot with compliant masts*," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4744–4750, IEEE, May 2024.
- [29] A. Lucarini, E. Idà, and M. Carricato, "Optimal design of a deployable and reconfigurable cable-driven parallel robot," in *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp. 1–6, IEEE, Sept. 2024.
- [30] Y. Hu, H. Liu, and H. Yuan, "A portable cable-suspended parallel robot and its applications in indoor inspection," *IEEE Robotics and Automation Letters*, vol. 9, pp. 10644–10651, Nov. 2024.
- [31] R. Verhoeven and M. Hiller, *Estimating the Controllable Workspace of Tendon-Based Stewart Platforms*, pp. 277–284. Springer Netherlands, 2000.
- [32] I. Ebert-Uphoff and P. Voglewede, "On the connections between cable-driven robots, parallel manipulators and grasping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, IEEE, 2004.
- [33] M. Gouttefarde, J.-P. Merlet, and D. Daney, "Wrench-feasible workspace of parallel cable-driven mechanisms," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, Apr. 2007.
- [34] A. Pott, T. Bruckmann, and L. Mikelsons, *Closed-form Force Distribution for Parallel Wire Robots*, pp. 25–34. Springer Berlin Heidelberg, 2009.
- [35] A. Pott, *An Improved Force Distribution Algorithm for Over-Constrained Cable-Driven Parallel Robots*, pp. 139–146. Springer Netherlands, Oct. 2013.

- [36] M. Gouttefarde, J. Lamaury, C. Reichert, and T. Bruckmann, "A versatile tension distribution algorithm for n -dof parallel robots driven by $n + 2$ cables," *IEEE Transactions on Robotics*, vol. 31, pp. 1444–1457, Dec 2015.
- [37] E. Picard, S. Caro, F. Plestan, and F. Claveau, *Stiffness Oriented Tension Distribution Algorithm for Cable-Driven Parallel Robots*, pp. 209–217. Springer International Publishing, July 2020.
- [38] H. Gao, G. Sun, Z. Liu, C. Sun, N. Li, L. Ding, H. Yu, and Z. Deng, "Tension distribution algorithm based on graphics with high computational efficiency and robust optimization for two-redundant cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 172, p. 104739, June 2022.
- [39] M. A. Khosravi and H. D. Taghirad, "Robust pid control of fully-constrained cable driven parallel robots," *Mechatronics*, vol. 24, no. 2, pp. 87–97, 2014.
- [40] Y. Majdi, H. Marufkhani, and M. A. Khosravi, "Active sliding mode fault tolerant control of a fully-constrained planar cable-driven parallel robot," in *2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM)*, IEEE, Dec. 2023.
- [41] J. C. Santos, A. Chemori, and M. Gouttefarde, "Model predictive control of large-dimension cable-driven parallel robots," in *Cable-Driven Parallel Robots* (A. Pott and T. Bruckmann, eds.), (Cham), pp. 221–232, Springer International Publishing, 2019.
- [42] J. C. Santos, M. Gouttefarde, and A. Chemori, "A nonlinear model predictive control for the position tracking of cable-driven parallel robots," *IEEE Transactions on Robotics*, pp. 1–20, 2022.
- [43] V. Mattioni, E. Idà, and M. Carricato, "Force-distribution sensitivity to cable-tension errors in overconstrained cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 175, p. 104940, 2022.
- [44] L. Guagliumi, A. Berti, E. Monti, M. Fabritius, C. Martin, and M. Carricato, "Force-sensor-free implementation of a hybrid position–force control for overconstrained cable-driven parallel robots," *Robotics*, vol. 13, p. 25, Jan. 2024.
- [45] V. Mattioni, E. Idà, M. Gouttefarde, and M. Carricato, "A practical approach for the hybrid joint-space control of overconstrained cable-driven parallel robots," in *Cable-Driven Parallel Robots* (S. Caro, A. Pott, and T. Bruckmann, eds.), (Cham), pp. 149–160, Springer Nature Switzerland, 2023.
- [46] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, *Control Stability Workspace for a Cable-Driven Parallel Robot Controlled by Visual Servoing*, pp. 284–296. Springer International Publishing, 2021.
- [47] E. Ida, F. Zoffoli, and M. Carricato, *Hybrid-Control-Based Workspace Analysis of Overconstrained Cable-Driven Parallel Robots*, pp. 324–331. Springer Nature Switzerland, 2024.

-
- [48] F. Zoffoli, E. Ida', and M. Carricato, "Design and control optimization for hybrid-controlled overconstrained cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 209, p. 105998, July 2025.
- [49] E. Ida' and M. Carricato, "Static workspace computation for underactuated cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 193, p. 105551, Mar. 2024.
- [50] S. Gabaldo, E. Idà, and M. Carricato, "Pose-estimation methods for underactuated cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 199, p. 105690, Sept. 2024.
- [51] E. Idà and V. Mattioni, "Cable-driven parallel robot actuators: State of the art and novel servo-winch concept," *Actuators*, vol. 11, p. 290, Oct. 2022.
- [52] F. Zoffoli, E. Idà, and M. Carricato, *Initial-Pose Self-calibration for Deployable Overconstrained Cable-Driven Parallel Robots*, pp. 559–567. Springer Nature Switzerland, 2025.
- [53] E. Idà, S. Briot, and M. Carricato, "Identification of the inertial parameters of underactuated cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 167, p. 104504, 2022.
- [54] H. Khalil, *Nonlinear Systems*. Pearson Education, Prentice Hall, 2002.
- [55] J. A. dit Sandretto, D. Daney, and M. Gouttefarde, *Calibration of a Fully-Constrained Parallel Cable-Driven Robot*, pp. 77–84. Springer Vienna, 2013.
- [56] D. Lau, *Initial Length and Pose Calibration for Cable-Driven Parallel Robots with Relative Length Feedback*, pp. 140–151. Springer International Publishing, July 2017.
- [57] B. Wang and S. Caro, *Exit Point, Initial Length and Pose Self-calibration Method for Cable-Driven Parallel Robots*, pp. 90–101. Springer International Publishing, 2021.
- [58] R. J. Caverly, K. Bunker, S. Patel, and V. L. Nguyen, *Forward Kinematics and On-line Self-calibration of Cable-Driven Parallel Robots with Covariance-Based Data Quality Assessment*, pp. 369–380. Springer Nature Switzerland, 2023.
- [59] R. J. Caverly, S. K. Cheah, K. R. Bunker, S. Patel, N. Sexton, and V. L. Nguyen, "Online self-calibration of cable-driven parallel robots using covariance-based data quality assessment metrics," *Journal of Mechanisms and Robotics*, vol. 17, June 2024.
- [60] F. Zoffoli, V. Coccia, E. Idà, and M. Carricato, *A Rapid Initial-Pose Self-calibration Method for Underactuated Cable-Driven Parallel Robots*, pp. 366–374. Springer Nature Switzerland, 2024.
- [61] F. Zoffoli, E. Idà, and M. Carricato, "A comparative analysis of initial-pose self-calibration algorithms for underactuated cable-driven parallel robots," *Robotica*, pp. 1–17, Oct. 2025.

- [62] E. Idá, J.-P. Merlet, and M. Carricato, *Automatic Self-Calibration of Suspended Under-Actuated Cable-Driven Parallel Robot using Incremental Measurements*, pp. 333–344. Springer International Publishing, 2019.
- [63] E. Idà, T. Bruckmann, and M. Carricato, “Rest-to-rest trajectory planning for underactuated cable-driven parallel robots,” *IEEE Transactions on Robotics*, vol. 35, pp. 1338–1351, Dec. 2019.
- [64] E. Ueland, T. Sauder, and R. Skjetne, “Optimal force allocation for overconstrained cable-driven parallel robots: Continuously differentiable solutions with assessment of computational efficiency,” *IEEE Transactions on Robotics*, vol. 37, pp. 659–666, Apr. 2021.
- [65] L. Gagliardini, M. Gouttefarde, and S. Caro, *Determination of a Dynamic Feasible Workspace for Cable-Driven Parallel Robots*, pp. 361–370. Springer International Publishing, July 2017.
- [66] M. Gouttefarde, J.-F. Collard, N. Riehl, and C. Baradat, “Geometry selection of a redundantly actuated cable-suspended parallel robot,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 501–510, 2015.
- [67] D. Song, L. Zhang, and F. Xue, “Configuration optimization and a tension distribution algorithm for cable-driven parallel robots,” *IEEE Access*, vol. 6, pp. 33928–33940, 2018.
- [68] C. Reichert and T. Bruckmann, “Optimization of the geometry of a cable-driven storage and retrieval system,” in *Robotics and Mechatronics* (R. (Chunhui) Yang, Y. Takeda, C. Zhang, and G. Fang, eds.), (Cham), pp. 225–237, Springer International Publishing, 2019.
- [69] L. Gagliardini, S. Caro, M. Gouttefarde, P. Wenger, and A. Girin, “Optimal design of cable-driven parallel robots for large industrial structures,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5744–5749, 2014.
- [70] L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin, “A reconfiguration strategy for reconfigurable cable-driven parallel robots,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1613–1620, 2015.
- [71] L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin, “Discrete reconfiguration planning for cable-driven parallel robots,” *Mechanism and Machine Theory*, vol. 100, pp. 313–337, June 2016.
- [72] L. Gagliardini, M. Gouttefarde, and S. Caro, “Design of reconfigurable cable-driven parallel robots,” in *Mechatronics for Cultural Heritage and Civil Engineering* (E. Ottaviano, A. Pelliccio, and V. Gattulli, eds.), pp. 85–113, Cham: Springer International Publishing, 2018.
- [73] Y. Peng and W. Bu, “Analysis of reachable workspace for spatial three-cable under-constrained suspended cable-driven parallel robots,” *Journal of Mechanisms and Robotics*, vol. 13, May 2021.

-
- [74] J. Pusey, A. Fattah, S. Agrawal, and E. Messina, "Design and workspace analysis of a 6-6 cable-suspended parallel robot," *Mechanism and Machine Theory*, vol. 39, pp. 761–778, July 2004.
- [75] E. Stump and V. Kumar, "Workspaces of cable-actuated parallel manipulators," *Journal of Mechanical Design*, vol. 128, pp. 159–167, July 2005.
- [76] M. Gouttefarde and C. Gosselin, "Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms," *IEEE Transactions on Robotics*, vol. 22, pp. 434–445, June 2006.
- [77] C. B. Pham, S. H. Yeo, G. Yang, and I.-M. Chen, "Workspace analysis of fully restrained cable-driven manipulators," *Robotics and Autonomous Systems*, vol. 57, pp. 901–912, Sept. 2009.
- [78] S. Perreault, P. Cardou, C. M. Gosselin, and M. J.-D. Otis, "Geometric Determination of the Interference-Free Constant-Orientation Workspace of Parallel Cable-Driven Mechanisms," *Journal of Mechanisms and Robotics*, vol. 2, p. 031016, 07 2010.
- [79] M. Gouttefarde, D. Daney, and J.-P. Merlet, "Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots," *IEEE Transactions on Robotics*, vol. 27, pp. 1–13, Feb. 2011.
- [80] R. Verhoeven and M. Hiller, *Tension Distribution in Tendon-Based Stewart Platforms*, pp. 117–124. Springer Netherlands, 2002.
- [81] M. Hassan and A. Khajepour, "Minimum-norm solution for the actuator forces in cable-based parallel manipulators based on convex optimization," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, Apr. 2007.
- [82] M. Hassan and A. Khajepour, "Minimization of bounded cable tensions in cable-based parallel manipulators," in *Volume 8: 31st Mechanisms and Robotics Conference, Parts A and B*, IDETC-CIE2007, ASMEDC, Jan. 2007.
- [83] T. Bruckmann, L. Mikelsons, M. Hiller, and D. Schramm, "A new force calculation algorithm for tendon-based parallel manipulators," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, IEEE, 2007.
- [84] L. Mikelsons, T. Bruckmann, M. Hiller, and D. Schramm, "A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, May 2008.
- [85] J. Lamaury and M. Gouttefarde, *A Tension Distribution Method with Improved Computational Efficiency*, pp. 71–85. Springer Berlin Heidelberg, Sept. 2012.
- [86] K. Müller, C. Reichert, and T. Bruckmann, *Analysis of a Real-Time Capable Cable Force Computation Method*, pp. 227–238. Springer International Publishing, Aug. 2014.

- [87] J. Lamaury and M. Gouttefarde, "Control of a large redundantly actuated cable-suspended parallel robot," in *2013 IEEE International Conference on Robotics and Automation*, IEEE, May 2013.
- [88] E. Idà, S. Briot, and M. Carricato, "Robust trajectory planning of under-actuated cable-driven parallel robot with 3 cables," in *Advances in Robot Kinematics 2020* (J. Lenarčič and B. Siciliano, eds.), (Cham), pp. 65–72, Springer International Publishing, 2021.
- [89] A. Pott, "Efficient computation of the workspace boundary, its properties and derivatives for cable-driven parallel robots," in *Computational Kinematics* (S. Zeghloul, L. Romdhane, and M. A. Laribi, eds.), (Cham), pp. 190–197, Springer International Publishing, 2018.
- [90] F. Zaccaria, E. Ida, S. Briot, and M. Carricato, "Workspace computation of planar continuum parallel robots," *IEEE Robotics and Automation Letters*, vol. 7, pp. 2700–2707, Apr. 2022.
- [91] F. Zaccaria, E. Idá, and S. Briot, "A boundary computation algorithm for the workspace evaluation of continuum parallel robots," *Journal of Mechanisms and Robotics*, vol. 16, June 2023.
- [92] S. Kawamura, H. Kino, and C. Won, "High-speed manipulation by using parallel wire-driven robots," *Robotica*, vol. 18, pp. 13–21, Jan. 2000.
- [93] M. Fabritius, G. Rubio-Gómez, C. Martin, J. C. Santos, W. Kraus, and A. Pott, "A nullspace-based force correction method to improve the dynamic performance of cable-driven parallel robots," *Mechanism and Machine Theory*, vol. 181, p. 105177, Mar. 2023.
- [94] J. C. Santos, A. Chemori, and M. Gouttefarde, "Redundancy resolution integrated model predictive control of cdprs: Concept, implementation and experiments," in *2020 IEEE International Conference on Robotics and Automation (ICRA), (ICRA) OR "International Conference on Robotics and Automation"*, pp. 3889–3895, 2020.
- [95] S. Bouchard and C. M. Gosselin, "A simple control strategy for overconstrained parallel cable mechanisms," in *Proceedings of the 20th Canadian Congress of Applied Mechanics, CANCAM2005*, p. 1540–1545, 30 May - 6 June, Montreal, Quebec, Canada, 2005.
- [96] G. Mottola, C. Gosselin, and M. Carricato, "Effect of actuation errors on a purely-translational spatial cable-driven parallel robot," in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 701–707, 2019.
- [97] P. Cardou, S. Bouchard, and C. Gosselin, "Kinematic-sensitivity indices for dimensionally nonhomogeneous jacobian matrices," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 166–173, 2010.
- [98] J. P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *Journal of Mechanical Design*, vol. 128, pp. 199–206, June 2005.

-
- [99] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes,” *ACM Transactions on Graphics*, vol. 13, pp. 43–72, Jan. 1994.
- [100] Z. Zhang, G. Xie, Z. Shao, and C. Gosselin, “Kinematic calibration of cable-driven parallel robots considering the pulley kinematics,” *Mechanism and Machine Theory*, vol. 169, p. 104648, Mar. 2022.
- [101] Z. Liu, Z. Qin, H. Gao, G. Sun, Z. Huang, and Z. Deng, “Initial-pose self-calibration for redundant cable-driven parallel robot using force sensors under hybrid joint-space control,” *IEEE Robotics and Automation Letters*, vol. 8, pp. 1367–1374, Mar. 2023.
- [102] S. Qian, K. Bao, B. Zi, and N. Wang, “Kinematic calibration of a cable-driven parallel robot for 3d printing,” *Sensors*, vol. 18, p. 2898, Sept. 2018.
- [103] M. Russo and X. Dong, “A calibration procedure for reconfigurable gough-stewart manipulators,” *Mechanism and Machine Theory*, vol. 152, p. 103920, Oct. 2020.
- [104] H. Mahnke and R. J. Caverly, *Fast and Reliable Iterative Cable-Driven Parallel Robot Forward Kinematics: A Quadratic Approximation Approach*, pp. 3–15. Springer Nature Switzerland, 2025.
- [105] S. Gabaldo, E. Idà, and M. Carricato, *Pose-Estimation Methods for Planar Under-actuated Cable-Driven Parallel Robots*, pp. 3–15. Springer Nature Switzerland, 2023.
- [106] C. Martin, M. Fabritius, J. T. Stoll, and A. Pott, “A laser-based direct cable length measurement sensor for cdprs,” *Robotics*, vol. 10, p. 60, Apr. 2021.
- [107] J.-H. Borm and C.-H. Meng, “Determination of optimal measurement configurations for robot calibration based on observability measure,” *The International Journal of Robotics Research*, vol. 10, pp. 51–63, Feb. 1991.
- [108] D. Daney, Y. Papegay, and B. Madeline, “Choosing measurement poses for robot calibration with the local convergence method and tabu search,” *The International Journal of Robotics Research*, vol. 24, pp. 501–518, June 2005.
- [109] F. Zhang, W. Shang, and S. Cong, “Choosing measurement configurations for kinematic calibration of cable-driven parallel robots,” in *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, IEEE, July 2018.
- [110] H. Wang, T. Gao, J. Kinugawa, and K. Kosuge, “Finding measurement configurations for accurate robot calibration: Validation with a cable-driven robot,” *IEEE Transactions on Robotics*, vol. 33, pp. 1156–1169, Oct. 2017.
- [111] A. Lucarini, E. Idà, and M. Carricato, *Rest-to-Rest Trajectory Planning for Under-actuated Cable-Driven Parallel Robots Using Input Shaping: Preliminary Results*, pp. 195–206. Springer Nature Switzerland, 2025.
- [112] P. Miermeister and A. Pott, *Auto Calibration Method for Cable-Driven Parallel Robots Using Force Sensors*, pp. 269–276. Springer Netherlands, 2012.

- [113] H. Zhuang, J. Wu, and W. Huang, "Optimal planning of robot calibration experiments by genetic algorithms," *Journal of Robotic Systems*, vol. 14, pp. 741–752, Oct. 1997.
- [114] H. Zhuang, J. Wu, and W. Huang, "Optimal planning of robot calibration experiments by genetic algorithms," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2 of *ROBOT-96*, pp. 981–986, IEEE, 1996.
- [115] A. Joubair and I. A. Bonev, "Comparison of the efficiency of five observability indices for robot calibration," *Mechanism and Machine Theory*, vol. 70, pp. 254–265, Dec. 2013.
- [116] Y. Sun and J. M. Hollerbach, "Observability index selection for robot calibration," in *2008 IEEE International Conference on Robotics and Automation*, pp. 831–836, IEEE, May 2008.
- [117] C. Huang, F. Xie, X.-J. Liu, and Q. Meng, "Measurement configuration optimization and kinematic calibration of a parallel robot," *Journal of Mechanisms and Robotics*, vol. 14, Dec. 2021.
- [118] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, July 1944.
- [119] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, pp. 431–441, June 1963.
- [120] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM Journal on Scientific Computing*, vol. 21, pp. 1–23, Jan. 1999.
- [121] J. A. Rice, *Mathematical statistics and data analysis*. Duxbury advanced series, Belmont, Calif.: Thomson Brooks/Cole, 3. ed., internat. student ed. ed., 2007. teilweise auch mit CD-ROM, ISBN der CD-ROM: 0-495-10986-X.
- [122] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. No. v.10 in New York Academy of Sciences Series, Hoboken, New Jersey: Wiley, third edition ed., 2017. Description based on publisher supplied metadata and other sources.
- [123] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 698–700, Sept. 1987.
- [124] D. Simon, *Optimal state estimation*. Hoboken, N.J: Wiley-Interscience, 2006.
- [125] R. Qi, M. Rushton, A. Khajepour, and W. W. Melek, "Decoupled modeling and model predictive control of a hybrid cable-driven robot (hcdr)," *Robotics and Autonomous Systems*, vol. 118, pp. 1–12, Aug. 2019.
- [126] M. H. Korayem, M. Yousefzadeh, and B. Beyranvand, "Dynamics and control of a 6-dof cable-driven parallel robot with visco-elastic cables in presence of measurement noise," *Journal of Intelligent & Robotic Systems*, vol. 88, pp. 73–95, Apr. 2017.

- [127] M. Korayem, M. Yousefzadeh, and S. Kian, "Precise end-effector pose estimation in spatial cable-driven parallel robots with elastic cables using a data fusion method," *Measurement*, vol. 130, pp. 177–190, Dec. 2018.
- [128] V. Le Nguyen and R. J. Caverly, "Cable-driven parallel robot pose estimation using extended kalman filtering with inertial payload measurements," *IEEE Robotics and Automation Letters*, vol. 6, pp. 3615–3622, Apr. 2021.
- [129] S. S. Haykin, ed., *Kalman filtering and neural networks*. Adaptive and learning systems for signal processing, communications, and control, New York: Wiley, 2001. Includes bibliographical references and index.