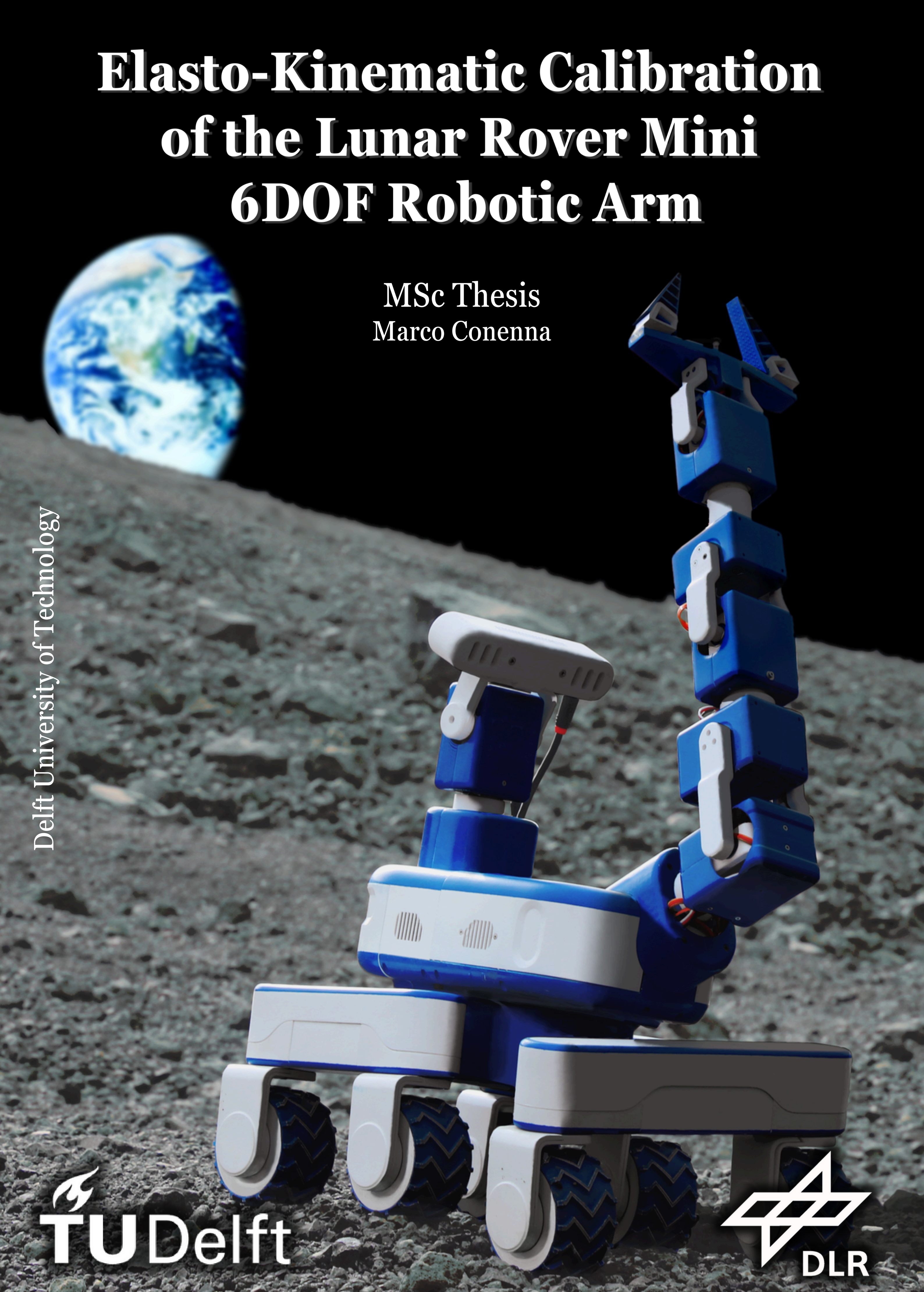
Elasto-Kinematic Calibration of the Lunar Rover Mini 6DOF Robotic Arm

MSc Thesis

Marco Conenna



Delft University of Technology

Elasto-Kinematic Calibration of the Lunar Rover Mini 6DOF Robotic Arm

by

Marco Conenna

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday November 20, 2024 at 13:45.

Student number:	5713536	
Project duration:	February 15, 2024 – November 20, 2024	
Thesis committee:	Dr. A. Menicucci	TU Delft, chair
	Dr. ir. E. van Kampen	TU Delft, independent examiner
	Dr. J. Guo,	TU Delft, supervisor
	Dr. Ing. A. Wedler,	DLR, supervisor

Cover image edit credits: G. Bussaglia [https://www.artstation.com/aldebubi]

TUDelft DLR

Preface

This Master's thesis is the product of a collaboration between TU Delft and the Robotics and Mechatronics Institute of the German Aerospace Center (DLR). Its central topic is the improvement of the performance of the robotic arm integrated into the Lunar Rover Mini, a small-scaled rover which serves as a learning platform for the next generations of students and researchers in the field of space robotics.

I would like to express my deepest gratitude to Dr. Jian Guo for supervising this project and to Dr. Ing. Armin Wedler for allowing me to join his team at DLR and work on such an engaging system, which I cared for as if it were my own (despite painting it in the wrong shade of blue, and accidentally let it roll down the crater of a volcano).
Special thanks also go to Riccardo, whose assistance was crucial in bringing the little rover to the Vulcano testing trip; to Peter Lehner, for his key suggestions on the methodology of this work; and to all my colleagues at DLR who provided invaluable support along the way.

Thank you Florian, Simon, Srujan, Oliver, and Pietro, for your emotional and technical support (not Florian, yours was only emotional, sorry) during the last 13 months at DLR. Just always remember to never do what you don't know, that life is what happens when you are busy making other plans, and that emus are not capable of driving F1 cars.

Moving forward from what many will find nonsense, I thank my Italian colleagues at TU Delft, the Olimpo. I will never forget the countless nights in the hall of the Student Hotel desperately writing assignments in your company, and our adventures culminating in the Oktoberfest 2023 incident.

A huge thank you to my Erasmus friends in Munich, who showed me that the true Master's degree is the friends made along the way and that TV jingles can hide dark secrets.

Thank you to my friends from Fano & surrounding areas for supporting me from 1432/720 km away and always making me feel welcome whenever I'm back home.

Last but most importantly, to my family: my mom, dad, my brother Andrea, and my beautiful golden retriever Murphy, for always being by my side and supporting me in all my adventures. Grazie!

Marco Conenna
Delft, November 2024

Contents

List of Figures

List of Tables

Acronyms

ASA Acrylonitrile Styrene Acrylate

CAD Computer Aided Design

CMM Coordinate Measuring Machine

DH Denavit Hartenberg

DOF Degrees of Freedom

FEA Finite Element Analysis

FK Forward Kinematics

IK Inverse Kinematics

LRM Lunar Rover Mini

LRU Lightweight Rover Unit

MDH Modified Denavit Hartenberg

MSA Matrix Structural Analysis

PLA Polylactic Acid

POE Product of Exponentials

PWM Pulse With Modulation

RC Radio Control

RMPL Robot Motion Planning Library

SVD Singular Values Decomposition

TCP Tool Center Point

VJM Virtual Joint Model

1

Introduction

Over the last 50 years, Rovers, uncrewed mobile robots, have played a pivotal role in humanity's endeavour in space exploration, becoming invaluable tools for traversing alien landscapes and gathering scientific data. Yet, their capabilities are often limited to observation and basic environmental interaction, and this is where the integration of robotic arms, also known as manipulators, takes place.

In 1976, the first rover equipped with a robotic arm landed on Mars, the Viking 1 (NASA, 1988), which allowed for the first time in history the in-situ analysis of extraterrestrial material, namely Martian soil. Approximately three decades later, advancements in robotic arm technology enabled a new era of complexity for Martian missions. The Pathfinder (NASA, 1997) and Sojourner (Matijevic, 1997) rovers leveraged these advanced manipulators to deploy and operate scientific instruments, significantly expanding the operational scope of rovers beyond basic observation. However, a substantial leap forward occurred in 2012 with the arrival of the Curiosity rover. Curiosity's sophisticated robotic arm, nicknamed ”The Arm”, (NASA, 2024) embodies a level of dexterity and precision unmatched by its predecessors. This remarkable feat of engineering empowers Curiosity to undertake intricate tasks such as rock drilling, sample collection, and in-depth analysis, fundamentally reshaping our understanding of what a space rover can achieve.

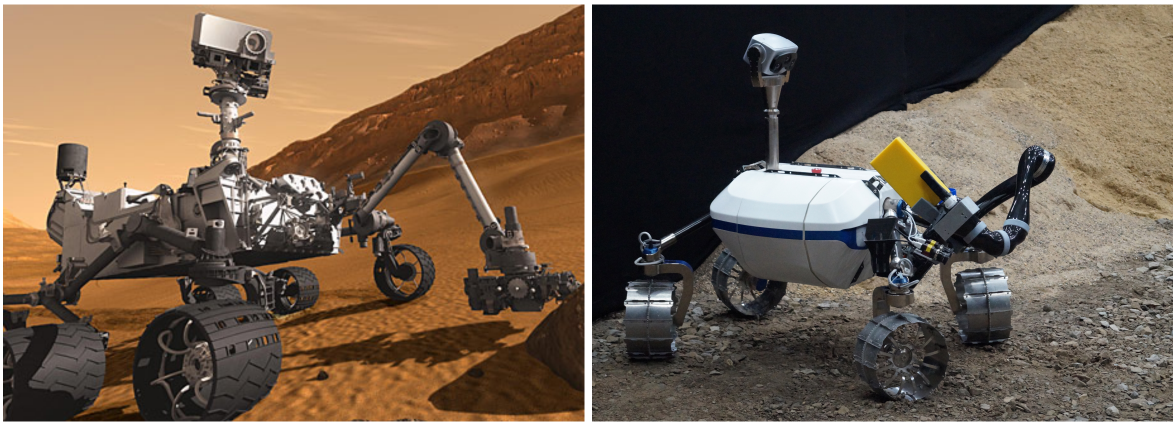


Figure 1.1: On the left, NASA's Mars Curiosity Rover and its advanced manipulator ”The Arm” (NASA, 2024). On the right DLR's Lightweight Rover Unit (LRU) while operating its extremely dextrous robotic arm (Wedler et al., 2015).

Looking ahead, even more powerful tools for grabbing and moving things in space will be key to future exploration, and this is where the expertise of institutions like the German Aerospace Center (DLR) comes into play. During the last decades, the DLR Robotics and Mechatronics Center has been a key player in developing advanced robotic technologies destined for space applications. Among the currently leading projects, the Lightweight Rover Unit (LRU) stands out for its cutting-edge individually steered wheels and the extremely precise and dextrous robotic arm mounted on its back (Wedler et al., 2015).

In addition, between 2017 and 2018, DLR's robot named ”Justin” participated in the METERON SUPVIS experiments in a collaboration between DLR and ESA, allowing astronauts to remotely control Justin's robotic arms from the International Space Station (DLR, 2023). This experiment once again proved how critical robotic manipulators will be in the near future, allowing humans to safely interact with objects placed thousands of kilometres from them. These delicate tasks, performed at such vast distances, demanded the utmost precision from Justin's arms. Even a minor misstep could have jeopardized the entire experiment, highlighting the paramount importance of precision in space exploration.

1.1. Lunar Rover Mini (LRM)

Taking inspiration from the LRU overall design and the peculiar 6-wheel kinematic chain developed in the institute for the aforementioned Exomars rover, in 2015 DLR started the design of a new system: Lunar Rover Mini (LRM) (Bekkers et al., 2023).

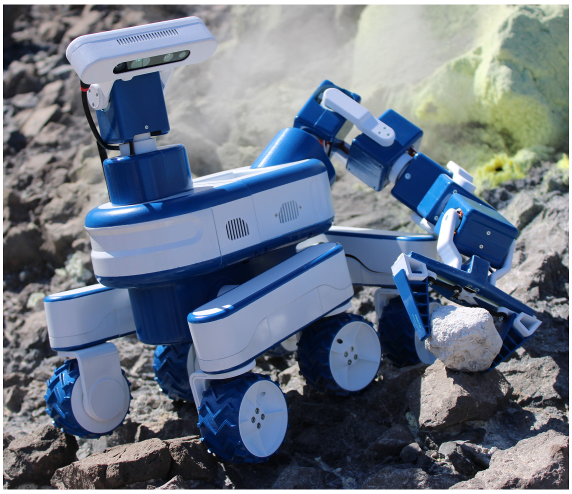


Figure 1.2: LRM in Vulcano, Italy.

The LRM project has grown steadily over the years, integrating student contributions and becoming part of various educational and research initiatives. The project is driven by the goal of making space exploration more accessible, not just to professional researchers but also to students and educators. By keeping the costs of these systems low, DLR aims to enable a wider audience to engage with space exploration challenges, particularly in areas such as cooperative robotics, where multiple robots must work together to map unknown terrains, localise themselves, and manage complex tasks. In 2023, the LRM rover exterior design was heavily updated.

Shown in figure 1.2, the scope of this rover is to enable students and researchers to conduct experiments on an inexpensive platform which shares the same complex high-level development stack of the costly rovers developed by the institute.

To achieve such a goal, the rover is only 36 × 26 × 39 cm (L×W×H) in size, and its entire structure is 3D-printed in Acrylonitrile Styrene Acrylate (ASA). Its main body is supported by a triple-bogie suspension system similar to the one employed on the Exomars rover, allowing the robot to adjust seamlessly to the irregular terrain beneath it. The rover hosts the following systems: the intel NUC, with an i7 processor running an Open-Suse Leap 15.4 operating system, is connected via USB cables to a dedicated PCB, which coordinates the communication between the main computer and each wheel. The pan-tilt unit used for autonomous navigation, is equipped with an Intel RealSense D435i camera moved by two Radio Control (RC) servos.

The camera allows the implementation of the open-source RTAB-Map module (Labbé & Michaud, 2018), which is fundamental for the main scope of the rover, which is mapping unknown territories by solving the SLAM problem (Simultaneous Localisation and Mapping). An example of this process in action is shown in Figure 1.3.

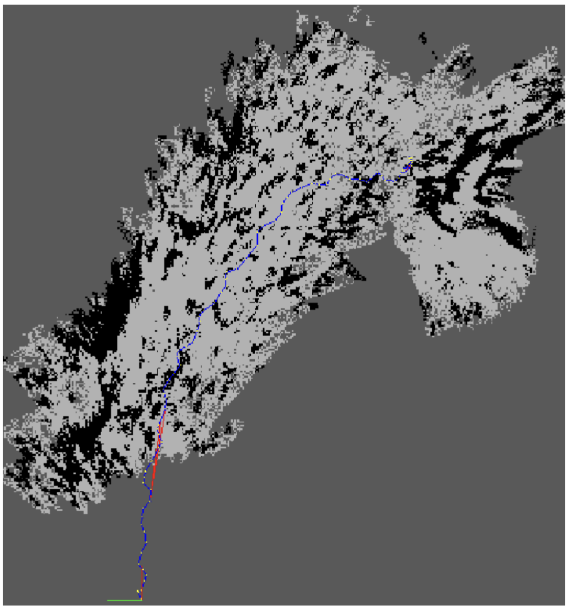


Figure 1.3: Occupancy grid of the Dried Moon Lake, Vulcano, Italy. Black areas are unavailable locations, light grey is free space and dark grey is unexplored area (Bekkers et al., 2023).

This module is backed up by an advanced semi-autonomous navigation protocol, which drives the rover towards the desired direction and detects and helps avoid obstacles. To facilitate the navigation, a particular wheel configuration allows the rover to proceed in three different driving modes: Ackermann, crabwalk, and simple rotation.

LRM shares the same high-level software implemented in LRU, which is based on the design principle of interoperability and modularity. A ”Links and Nodes Manager” (Schmidt, 2023) allows deploying a complex set of processes. Then, through a graphical user interface, the manager enables users to define the execution order of processes, monitor their status, and facilitate inter-process and real-time communication. In the context of the LRM, links and nodes allow the management of a diverse set of processes, which can be written in various programming languages, for example, python, c++ and Matlab.

The data flow is managed by Links and Nodes, and engages ROS (Robot Operating System) Melodic to facilitate the employment of perception, navigation and mapping functions.

1.2. LRM Robotic Manipulator: Preliminary Problem Statement

Until the end of 2023, LRM lacked a sample retrieval system, which limited its potential to research in fields related to the navigation problem (Bekkers, 2024), its advanced kinematics, and the sensors on board. For this reason, a group of students designed a robotic arm to be integrated into the rover, also aligning it more closely with the complex LRU.

The challenge represented by the rover’s strict weight and cost requirements and the need for a complex system with research potential led to a six Degrees of Freedom (DOF) robotic arm, entirely 3D printed and using inexpensive RC servos in each joint.

It is essential to highlight that the simplicity of this design, shown in figure 1.5, presents some critical features which must be addressed in the early phases of its integration with the rover. Preferring RC servos lacking a proper feedback system over heavier and more expensive motors similar to those used for the bogie wheels, as well as the total absence of metal components, inevitably adds

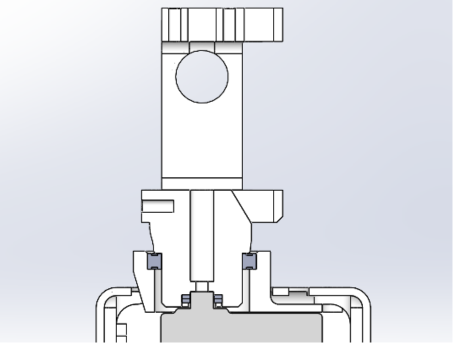


Figure 1.4: Cross-section of the area between the third and fourth joints, showing how the bearing system holds the vertical joints in place.

complexity to the integration process, as the arm presents a non-neglectable deformation even when unloaded. As this macroscopic error in the position of the end-effector caused by the flexibility of its components would undoubtedly impact the operability of the manipulator, this thesis aims to tackle this issue and deliver a fully functional and integrated robotic arm for LRM.

Considering the different loads each joint is expected to sustain, it was decided to use bigger servos for the three joints closer to the base and smaller ones for the last three.

To accommodate the varying loads each joint in the 6-DOF robotic arm will need to handle, larger servos were selected for the three joints closest to the base, while smaller servos were chosen for the last three joints. To ensure a smooth startup when the arm is powered on, the AGF High Torque series servos were selected, specifically, the A80BHMF for the larger joints and the A35BHL for the smaller ones. These servos are connected via robust ASA 3D-printed components, which have been meticulously designed to minimize play and maximize joint precision. The servos are attached to the plastic parts using metal heads that fit precisely with the servo's metal head. These metal heads are then securely encased in specific slots within the plastic components and further reinforced with an adhesive that effectively bonds steel to ASA plastic. The servo is firmly held in place by screwing the head into the plastic component. For joints 2, 4, and 6, this connection is achieved using vertical plastic parts that allow cable management. However, as illustrated in Figure 1.4, the configuration is more complex for joints 1, 3, and 5. In the figure, the bearing, depicted in dark grey, is integral to the joint's assembly. It securely holds the servo shaft, ensuring smooth rotational movement while maintaining alignment within the ASA 3D-printed housing. This setup minimizes play and keeps the joint stable under load.

Moreover, it must be noted that a particular assembly procedure must be followed to ensure the correct alignment between the parts, and avoid macroscopic errors. First, considering the steel heads that must be inserted in the ASA parts have three possible mounting options, and the number of teeth on the servo is 25, only one of those options is the correct one. When a servo must be mounted, it is necessary to power it up and move it to the home position first. Then, the plastic component, on which the steel head was already inserted, must be carefully fit on the servo head, and secured in place with a screw.

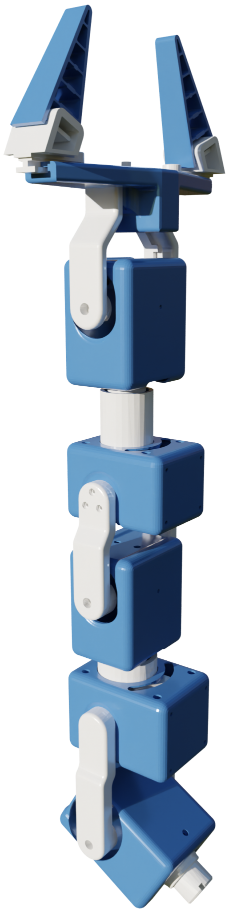


Figure 1.5: Render of the LRM Robotic Arm.

The limitations of the robotic arm stem from its reliance on inexpensive servos and its entirely 3D-printed construction, which compromise its overall accuracy and performance due to the deflection in the joints and links under load. Therefore, the primary objective of this thesis is to enhance the precision of the robotic arm while integrating it into the software architecture of the LRM to ensure it operates effectively within the intended applications.

To achieve this goal, the upcoming chapter will explore the current state of the art in elastic compensation techniques for robotic arms and present various methodologies that can be employed to tackle the challenges associated with joint elasticity and kinematic inaccuracies.

2

Literature Study: Elasto-Kinematic Calibration of Robotic Arms

This chapter will introduce the general mathematical background needed to understand the behaviour of robotic arms focussing on both forward and inverse kinematics, as well as the state-of-the-art in the field of robotics calibration. The three calibration levels will be described, with a particular focus on the third level which takes into account the main topic of this thesis project: the compensation for the links and joints elasticity.

2.1. Forward Kinematics

Forward, or direct kinematics, is a fundamental concept in robotics that establishes the connection between a robot's joint configuration and the resulting position and orientation of its end effector.

Siciliano et al. (2009) states that for calibration purposes, a kinematic model must respect the following basic requirements: completeness, continuity, and minimality. This means that the model should include enough parameters to describe any possible deviations of the actual kinematic parameters from the nominal values, that small physical deviations correspond to small changes in the parameters, and that these parameters should not be redundant.

This section will provide a brief mathematical background of how the state of the art addresses this problem, presenting the two main approaches: the Denavit Hartenberg (DH) convention and the Product of Exponentials (POE).

Additional approaches, such as the application of dual quaternions by G. Li et al. (2019), and the implementation of screw theory Choi et al. (2016), will not be addressed due to their complexity and the specificity of the contexts in which they were employed, deviating from the object of this thesis.

2.1.1. The Denavit-Hartenberg (DH) Convention

The first step in operating a robotic arm and predicting the end-effector's position is mathematically describing its behaviour. Among the several mathematical descriptions existing in modern literature, the one that stands out is the DH convention (Siciliano et al., 2009). In order to describe three-dimensionally the position of each joint with respect to a reference frame, in 1955 Jacques Denavit and Richard Hartenberg employed homogenous transformations to compactly represent the relationship between the coordinates of the same point in two different frames, giving a unique definition of each link frame. By carefully choosing the right reference frames for each link in accordance with the DH convention, the forward kinematics of a robotic arm can be computed as:

$$\mathbf{FK}(\mathbf{q}) = \mathbf{T}_1^0(q_1) \dots \mathbf{T}_n^{n-1}(q_n) = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & \mathbf{P}(\mathbf{q}) \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

where $\mathbf{R}(\mathbf{q})$ and $\mathbf{P}(\mathbf{q})$ are respectively the orientation and the position of the end effector with respect to Frame 0, and $\mathbf{T}_i^{i-1}$ are the homogenous transformation matrices which translate and rotate each reference frame with respect to the previous one. These matrices, which are function only of the joint variable q_i, that is θ_i for the revolute joint or d_i for a prismatic joint, take the form:

$$\mathbf{T}_i^{i-1}(q_i) = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where θ_i, α_i, a_i, and d_i, are the DH parameters, which strictly depend on the design of the robotic arm. θ_i represents the joint angle of the i^{th} joint, so a rotation about the z-axis of the $(i-1)^{th}$ frame; α_i is the twist angle between the i^{th} and $(i-1)^{th}$ frame, and represents a rotation about the common normal axis (the axis perpendicular to both the z-axes of the adjacent frames); a_i is the offset distance along the common normal axis between the z-axes of the i^{th} and $(i-1)^{th}$ frames; d_i represents the offset distance along the z-axis of the $(i-1)^{th}$ frame between the origins of the i^{th} and $(i-1)^{th}$ frames.

The next step in this convention is defining the body Jacobian matrix $\mathbf{J}_{body}$ of the manipulator (Kana et al., 2022), which expresses a relation between the joints and the end-effector velocities, can be defined as:

$$\mathbf{J}_{body} = [\mathbf{Ad}^{-1}_{T^1_E}\boldsymbol{\xi}_{b1}, \mathbf{Ad}^{-1}_{T^2_E}\boldsymbol{\xi}_{b2}, \dots, \mathbf{Ad}^{-1}_{T^n_E}\boldsymbol{\xi}_{bn}] \tag{2.3}$$

where $\boldsymbol{\xi}_{bn} = (\mathbf{T}_n^{n-1})^{-1} \cdot \partial_{q_n}\mathbf{T}_n^{n-1}$ and $\mathbf{Ad}_{\mathbf{T}^n_E}$ is the adjoint matrix for the homogenous transformation of the end-effector frame to the n^{th} joint frame.
The body Jacobian can be also written as:

$$\mathbf{J}_{body} = \begin{bmatrix} \mathbf{J}_{pos} \\ \mathbf{J}_{rot} \end{bmatrix} \tag{2.4}$$

where $\mathbf{J}_{pos}$ and $\mathbf{J}_{rot} \in \mathbb{R}^{3\times N}$ are respectively the top and bottom submatrices.

Although this convention is undoubtedly the most popular in models that present parallel revolute axes, it must be noted that it does not respect the second rule of Siciliano et al. (2009), as it faces the issue of singularities. A simple solution is adopting the ”Modified DH Convention”, documented in the work of Hayati (1983), which consists of adding extra translation and rotation matrices to the chain of homogenous transformations.

2.1.2. Product of Exponentials (POE)

Brockett (1984) devised the first alternative description to the DH convention to describe the forward kinematics of robotic arms. Named the Product of Exponentials, or POE, it still consists of a chain of transformations linking the zero reference frame to the end-effector, similar to the DH convention but with a completely different syntax, using affine (or adjoint) instead of homogenous transformations.
The following method is used to parametrize the affine transformation matrix between the base and the end-effector frames. Firstly, the ”zero configuration” is set, obtaining the following 4x4

matrix $g_{st}(0)$ linking the base frame to the tool frame when all the joints are set to the initial position:

$$g_{st}(0) = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{2.5}$$

where R is a 3x3 rotation matrix, and p the 1x3 translation vector. Then, for each joint of the kinematic chain, an origin point q and an axis of action are selected for the zero configuration with respect to the base reference frame. In the case of a prismatic joint the axis of action v is the vector along which the joint extends, while in the case of a revolute joint, the axis of action ω is the vector normal to the rotation.
The following twist vectors describe the movement of each joint, in order a revolute and a prismatic joint:

$$\xi_i = \begin{pmatrix} -\omega_i \times q_i \\ \omega_i \end{pmatrix} \qquad \xi_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix} \tag{2.6}$$

The resulting twist can be expressed as two 1x3 vector components, linear motion along an axis v and rotational motion along the same axis ω:

$$\xi = \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{2.7}$$

Rewriting The 3x1 vector ω in cross product matrix notation:

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.8}$$

Applying Rodrigues' rotation formula, the rotation formula is then calculated:

$$e^{\hat{\omega}\theta} = I + \hat{\omega} sin(\theta) + \hat{\omega}^2(1 - cos(\theta)) \tag{2.9}$$

As well as the translation formula:

$$t = (I - e^{\hat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \tag{2.10}$$

where I is the 3x3 identity matrix.
Combining the results for each joint, the following complete 4x4 matrix is obtained:

$$e^{\hat{\xi}_i\theta_i} = \begin{bmatrix} e^{\hat{\omega}\theta} & t \\ 0 & 1 \end{bmatrix} \tag{2.11}$$

In order to produce the affine transformation from the base frame to the end-effector, all the matrix exponentials must be multiplied:

$$g_d = e^{\hat{\xi}_1\theta_1} ... e^{\hat{\xi}_n\theta_n} g_{st}(0) \tag{2.12}$$

2.2. Inverse Kinematics

Inverse Kinematics (IK) uses the kinematics equation to predict the position of each joint of the robotic arm to reach a desired end-effector pose. The main difference with forward kinematics is that unless the robotic system is extremely simple, the solution to the IK problem is not unique and, therefore, needs iterative methods to be solved, specifically gradient-based optimisation methods.

The most notorious methods employed for solving the IK problem are listed below and retrieved from the work of Chin et al. (1997):

- **Newton's method for Non-Linear Systems**
  Newton's method is simple and converges only if the initial guess is close to the minimum. Each new joint angle at iteration k is calculated as:

$$\theta_{k+1} = \theta_k + \dot{\theta}_k \tag{2.13}$$

$$\dot{\theta}_k = \mathbf{J}^{-1} \times \mathbf{F}(\mathbf{X}_k) \tag{2.14}$$

  where $\mathbf{J}$ is the Jacobian Matrix that must be square and non-singular. $\mathbf{F}(\mathbf{X}_k)$ is the objective function;

- **Broyden-Fletcher-Goldfarb-Shanno (BFGS)**
  The BFGS gradient projection algorithm is a quasi-Newton method that leverages gradients from previous iterations to approximate second-order derivative information. These approximations aid in determining the step to take in the current iteration. Employing a gradient projection technique, it tackles boundary constraints posed by the joint limits of the robot model on the cost function, ensuring that the calculated direction is consistently adjusted to maintain validity in the search direction;

- **Fletcher-Reeves-Polak-Ribiere Method (FRPRM)**
  The Fletcher-Reeves-Polak-Ribiere Method (FRPRM) makes use of conjugate gradient vectors, which are calculated from gradient evaluations of the objective function. The step length λ_k which minimizes the objective function is calculated according to:

$$\lambda_k = (g_k \times g_{k+1}) g_k^{-2} \tag{2.15}$$

  where g_k is the gradient of the objective function at iteration k. This method converges quickly when the solution is near the local minimum;

- **Powell's method**
  Powell's method does not require any derivative to be calculated, as it proceeds by traversing a given search direction vector and updating each element in this vector iteratively. It does not converge when the search directions become linearly dependent.

- **PRAXIS method**
  The PRAXIS method is similar to Powell's method, but resetting the elements of the direction vector after every N iterations, it eliminates the linear dependence of the search directions and the potential lack of convergence possible with the former method;

- **Levenberg-Marquardt Algorithm**
  The Levenberg-Marquardt Algorithm is an error-damped least-squares method where the error-damped factor helps to prevent the algorithm from escaping a local minimum (Sugihara, 2011). This algorithm is optimised to converge much faster if the initial guess

is close to the solution, and for this reason, it is advisable to be used when the goal is finding IK solutions for a series of poses along a desired trajectory of the end effector. Once a robot configuration is found for one pose, that configuration is often a good initial guess at an IK solution for the next pose in the trajectory.

In modern commercially available tools, two methods are mainly implemented out of these lists: the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Projection Algorithm for singular poses, and the Levenberg-Marquardt Algorithm for trajectories.

2.3. Calibration

Once a robotic system's forward and inverse kinematics are complete, the next step to render it operational and ensure high precision when predicting the position of its components is calibration. This process is necessary due to several factors that may influence the accuracy of such systems. Karan and Vukobratović (1994) lists them into six categories: environmental (for instance temperature changes), parametric(variation of kinematic parameters, joint zero-reference offsets, influence of dynamic parameters, drive-train compliance, friction and other nonlinearities, including hysteresis and backlash), measurement (resolution and nonlinearity of joint position sensors), computational (computer round-off and steady-state servo errors), and finally application errors (installation errors, and workpiece position and geometry errors). While each error source can be addressed individually, it is advantageous to categorize them into three calibration levels and, for each level, define the modelling, measurement, identification and compensation steps. The following subsections describe the definition of these three levels which Elatta et al. (2003) and Roth et al. (1987) proposed in their detailed examination of robot calibration.

2.3.1. Level 1: Joint Linear Displacement

Often dealt with during the robot's construction, this level ensures the reading from a joint sensor corresponds to the correct joint displacement.

- **Model**
  It is the simple relationship between the output of the joint sensor and the actual joint displacement, often assumed to be linear. It can be written as:

$$\theta_i = \kappa_{i1}\eta_i + \kappa_{i2} \tag{2.16}$$

  where θ_i is the actual joint displacement, η_i is the signal from the transducer, and the vector $[\kappa_{i1}\eta_i, \kappa_{i2}]^T$ contains the calibration parameters that must be determined. In cases where higher precision is needed, nonlinear effects such as gear backlash or joint clearance may be included in the model if linearized. When this is not possible, they must be considered in level 3 calibration;

- **Measurement**
  It consists of employing an external measurement device, or moving the joints to some "known" configuration;

- **Identification**
  the constants in $[\kappa_{i1}\eta_i, \kappa_{i2}]^T$ can be easily determined by placing each joint at two known displacements, reading the transducer signal and then interpolating the values;

- **Correction**
  The conversion can be easily implemented in the control code or through specialized circuitry.

2.3.2. Level 2: Kinematic Model Calibration

The aim of level 2 calibration is to enhance the precision of both the manipulator's kinematic model and the correlation between joint transducers and actual joint displacement. This entails encompassing level 1 calibration procedures within level 2. During level 2 calibration, the robot's links are presumed to be rigid, and the joints are assumed to be flawless, allowing no unintended motion about their axes. Consequently, compensating for errors arising from joint clearances, for instance, is not part of level 2 calibration. The primary objective of level 2 calibration is to ascertain the spatial kinematic relationship between the joints and links.

At this level, the model is not limited to the joints and their relative displacement; an actual kinematic model is needed. From table 2.1, which shows a brief comparison of the state-of-the-art calibration techniques, it is clear that POE and the DH convention are the two preferred options. It must be noted that due to the singularity problem, the error models based on the DH methods are not continuous when two consecutive joint axes are close to being parallel, and therefore, for complex applications, the POE description is often preferred (Hayati, 1983).

- **Denavit-Hartenberg Convention**

  – Model

  Following the work of Kana et al. (2022), the ideal forward kinematics of an N-joint serial manipulator can be written as:

$$\mathbf{FK_p}(\mathbf{q}) = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & \mathbf{P_p}(\mathbf{q}) \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

  Where $\mathbf{R}$ and $\mathbf{P}$ are, respectively, the orientation and position of the end-effector, with subscript $\mathbf{p}$ representing the ideal link lengths retrieved from the Computer Aided Design (CAD) model of the arm.

  This calibration level aims to compensate for the assembly and manufacturing imprecisions, so it takes into account both the zero joint offsets, or angular offsets ($\delta\mathbf{q} = [q_1, ..., q_N]$), and the possible deviations of the link dimensions, or linear offsets ($\delta\mathbf{p} = [\mathbf{p}_1, ..., \mathbf{p}_{N+1}]$).

  Accounting for these errors, the transformation between two consecutive joint frames becomes:

$$\mathbf{T}_n^{n-1} = \begin{bmatrix} \mathbf{R}(q_n + \delta q_n) & \mathbf{p}_n + \delta\mathbf{p}_n \\ 0 & 1 \end{bmatrix} \tag{2.18}$$

  where $\mathbf{p}_n \in \mathbb{R}^{3\times1}$ is the ideal position vector of joint frame n with respect to joint frame (n-1) and $\delta\mathbf{p}_n \in \mathbb{R}^{3\times1}$ is the vector of linear offsets for the (n-1)th joint.

  Integrating these equations, the following notation for the corrected forward kinematics can be formulated:

$$\mathbf{FK_{p+\delta p}}(\mathbf{q} + \delta\mathbf{q}) = \begin{bmatrix} \mathbf{R}(\mathbf{q} + \delta\mathbf{q}) & \mathbf{P}_{\mathbf{p}_n+\delta\mathbf{p}_n}(\mathbf{q} + \delta\mathbf{q}) \\ 0 & 1 \end{bmatrix} \tag{2.19}$$

  – Measurement

  This phase can be carried out by employing different instruments. Subedi et al. (2022) provides the following options: Inertial Measurement Units (IMU), Laser Trackers, Theodolites, and ultrasonic sensors. Reading other works, it is also possible to use the feedback from the robot controller (Kana et al., 2022), a more precise

and already calibrated measurement arm (Wan et al., 2018), or Coordinate Measuring Machine (CMM) (Joubair & Bonev, 2014).
It must be noted that the measurements must be taken in particular poses to avoid occultation issues that often occur when small-sized trackers are implemented into complex systems. To overcome this issue, several studies take into account a preliminary analysis of the arm workspace, such as the works of Kolyubin et al. (2018), Sun and Hollerbach (2008), and Daney et al. (2005).

– Identification

Assuming the measured values sufficiently accurate, the ground truth can be noted as:

$$\widetilde{\mathbf{FK}}(\mathbf{q}) = \begin{bmatrix} \widetilde{\mathbf{R}}(\mathbf{q}) & \widetilde{\mathbf{P}}(\mathbf{q}) \\ 0 & 1 \end{bmatrix} \tag{2.20}$$

The goal is to choose $\delta\mathbf{q}$ and $\delta\mathbf{p}$ so that the difference between $\widetilde{\mathbf{FK}}(\mathbf{q})$ and $\mathbf{FK}_{\mathbf{p}+\delta\mathbf{p}}(\mathbf{q}+\delta\mathbf{q})$ is minimized.
The problem of finding the right $\delta\mathbf{q}$ and $\delta\mathbf{p}$ represents a non-linear regression problem, however, with the assumption that the orientation of the robotic arm does not depend on the linear offsets, the problem can be separated into two linear regression problems.

In order to identify the angular offsets $\delta\mathbf{q}$, Kana et al. (2022) and Kolyubin et al. (2018) adopt the Jacobian Matrix defined in Eq 2.3. As a first-order approximation, $\mathbf{R}(\mathbf{q}+\delta\mathbf{q})$ can be written as $\mathbf{R}(\mathbf{q}) + \frac{\partial\mathbf{R}(\mathbf{q})}{\partial\mathbf{q}}\delta\mathbf{q}$ therefore:

$$\mathbf{R}(\mathbf{q}) + \frac{\partial\mathbf{R}(\mathbf{q})}{\partial\mathbf{q}}\delta\mathbf{q} \approx \widetilde{\mathbf{R}}(q) \tag{2.21}$$

Substituting into this equation the rotation submatrix $\mathbf{J}_{rot}$ of the Body Jacobian Matrix, an optimal estimate of $\delta\mathbf{q}$ can be found via linear regression for the following system:

$$\begin{bmatrix} \mathbf{J}_{rot}(\mathbf{q}_1) \\ \vdots \\ \mathbf{J}_{rot}(\mathbf{q}_n) \end{bmatrix} \delta\mathbf{q} = \begin{bmatrix} (\mathbf{R}_1^T\widetilde{\mathbf{R}}_1 - \mathbb{I}_3)^V \\ \vdots \\ \mathbf{R}_M^T\widetilde{\mathbf{R}}_M - \mathbb{I}_3)^V \end{bmatrix} \tag{2.22}$$

where M is the number of measurements, the $(\cdot)^V$ operator turns 3×3 skew-symmetric matrices into corresponding 3D vectors, and $\mathbf{R_m} \equiv \mathbf{R}(\mathbf{q_m})$ and $\widetilde{\mathbf{R}}_\mathbf{m} \equiv \widetilde{\mathbf{R}}(\mathbf{q_m})$.

Once the $\delta\mathbf{q}$ have been found, the angle $\mathbf{q}'$ in the following equations will represent the corrected angle.
To find the linear offsets $\delta\mathbf{p}$ that minimize the difference:

$$\mathbf{P}_{\mathbf{p}+\delta\mathbf{p}}(\mathbf{q}'_\mathbf{m}) - \widetilde{\mathbf{P}}(\mathbf{q_m}) \tag{2.23}$$

The first-order approximation of the error can be defined as:

$$\mathbf{P}_{\mathbf{p}+\delta\mathbf{p}}(\mathbf{q}'_\mathbf{m}) - \widetilde{\mathbf{P}}(\mathbf{q_m}) \approx \mathbf{P_p}(\mathbf{q}'_\mathbf{m}) - \mathbf{P}(\mathbf{q_m}) + \nabla_\mathbf{p}\mathbf{P}\delta\mathbf{p} \tag{2.24}$$

Therefore, the optimal estimate of the linear offsets $\boldsymbol{\delta}\mathbf{p}$ can be computer as a linear regression of the following system:

$$\begin{bmatrix} \nabla_{\mathbf{p}}\mathbf{P}(\mathbf{q}'_1) \\ \vdots \\ \nabla_{\mathbf{p}}\mathbf{P}(\mathbf{q}'_\mathbf{M}) \end{bmatrix} \delta\mathbf{p} = \begin{bmatrix} \widetilde{\mathbf{P}}(\mathbf{q}_1) - \mathbf{P}_{\mathbf{p}}(\mathbf{q}'_1) \\ \vdots \\ \widetilde{\mathbf{P}}(\mathbf{q}_\mathbf{M}) - \mathbf{P}_{\mathbf{p}}(\mathbf{q}'_\mathbf{M}) \end{bmatrix} \tag{2.25}$$

– Correction

The correction consists of adding the following correct values of joint and link offsets $q'\ p'$:

$$\mathbf{q}' := \mathbf{q} + \boldsymbol{\delta}\mathbf{q} \tag{2.26}$$

$$\mathbf{p}' := \mathbf{p} + \boldsymbol{\delta}\mathbf{p} \tag{2.27}$$

to the arm's forward kinematics, which becomes:

$$\mathbf{FK}_{\mathbf{p}'}(\mathbf{q}') = \begin{bmatrix} \mathbf{R}(\mathbf{q}') & \mathbf{P}_{\mathbf{p}'}(\mathbf{q}') \\ 0 & 1 \end{bmatrix} \tag{2.28}$$

- **Product of Exponentials**

– Model

Equation 2.12 is re-written with a slightly different notation, taken from the work of Wu et al. (2015):

$$g = \prod_{i=1}^{n} exp(\hat{\boldsymbol{\xi}}_\mathbf{i} q_i) exp(\hat{\boldsymbol{\xi}}_\mathbf{st}) \tag{2.29}$$

where g is the transformation from the base frame to the tool frame, $\hat{\boldsymbol{\xi}}_\mathbf{i} \in se(3)$ (i = 1, 2, ..., N) is the n^{th} joint twist, and $\hat{\boldsymbol{\xi}_\mathbf{st}}$ is an initial twist determined by the initial configuration of the robot as well as the selection of the base frame and the tool frame.
By differentiating Eq 2.29 and right multiplying the inverse of g, the problem can be summarized into the minimization of the following cost function:

$$\underset{\delta\eta,\delta q,\delta\xi}{arg\ min} = \left\| \delta g g^{-1} - \left(\frac{\partial g}{\partial \xi} \frac{\partial \xi}{\partial \eta} \delta\eta + \frac{\partial g}{\partial \mathbf{q}} \delta\mathbf{q} + \frac{\partial g}{\partial \xi_{st}} \delta\xi_{st} \right) g^{-1} \right\|^2 \tag{2.30}$$

where $\delta\eta$ is a group of independent parameters describing the error of each joint twist. The traditional model by He et al. (2010) uses $\frac{\partial g}{\partial \xi}\delta\xi$ instead of $\frac{\partial g}{\partial \xi}\frac{\partial \xi}{\partial \eta}\delta\eta$, although $\delta\xi$ is redundant by definition. By linearizing the minimization problem, the following equation is obtained:

$$\mathbf{y} = \mathbf{Jx} \tag{2.31}$$

where

$$\mathbf{y} = [\delta g g^{-}1]^{V} \in \mathbb{R}^6 \tag{2.32}$$

$$\mathbf{J} = [\mathbf{J}_1, \mathbf{J}_2, ..., \mathbf{J}_n, \mathbf{J}_{st}] \in \mathbb{R}^{6\times(5r+3p+6)} \tag{2.33}$$

$$\mathbf{x} = [\delta\boldsymbol{\eta}_1^T, \delta q_1, ..., \delta\boldsymbol{\eta}_n^T, \delta q_n, \delta\boldsymbol{\xi}_{st}^T]^T \in \mathbb{R}^{5r+3p+6} \tag{2.34}$$

with $\mathbf{y}$ as the difference between the measured poses of the end-effector and the poses calculated with the nominal parameters, $\mathbf{J}$ the identification matrix, and $\mathbf{x}$ the parameters to be identified. r and p are, respectively, the numbers of revolute and prismatic joints.

- Measurement

  The tools applicable for this stage are detailed in the section discussing the DH convention. However, it's important to highlight that the literature outlines two main measurement methods: one focusing on both the position and orientation of the end-effector (He et al., 2010), and another solely on its position (Wu et al., 2015).

- Identification

  Running a mathematical identifiability inspection, He et al. (2010) states that $\delta\mathbf{q}$, $\delta\xi_{st}$, and $\delta\xi$ are identifiable only if any coordinate of joint twist cannot be the linear combination of the others, setting hard requirements for the poses the arm can be measured in. The last step is the identification of the parameters through an iterative least-squares algorithm. The procedure followed by Wu et al. (2015) using only position measurements is very similar, and both schemes are presented in Figure 2.1.

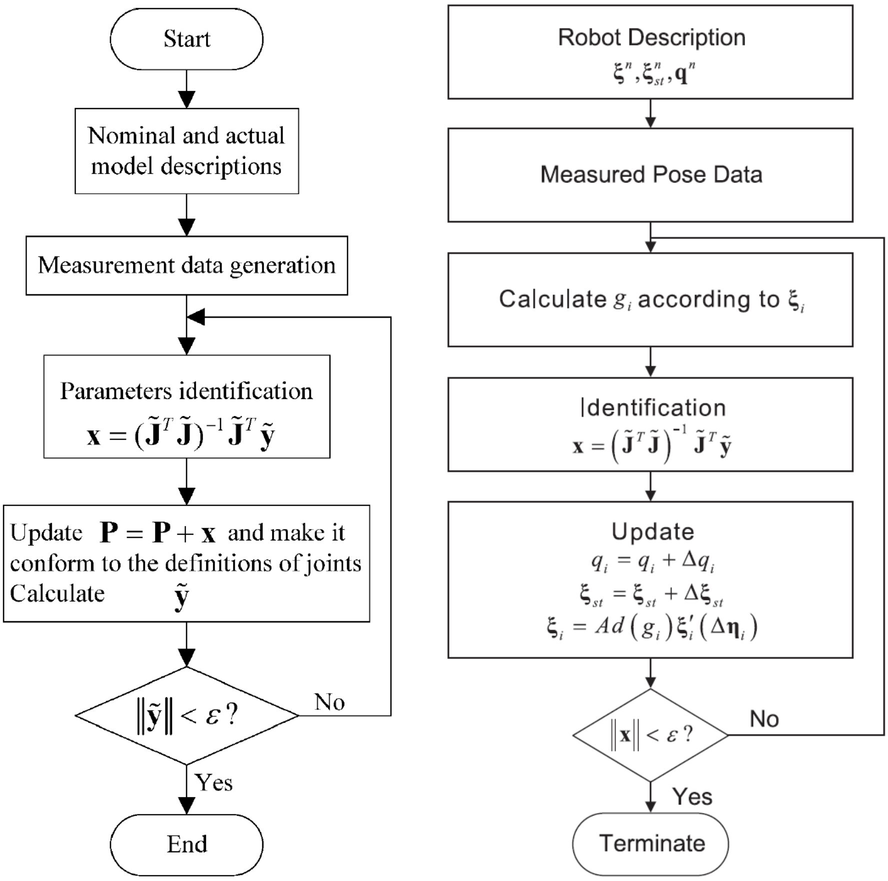

Start
Nominal and actual model descriptions
Measurement data generation
Parameters identification $\mathbf{x} = (\tilde{\mathbf{J}}^T\tilde{\mathbf{J}})^{-1}\tilde{\mathbf{J}}^T\tilde{\mathbf{y}}$
Update $\mathbf{P} = \mathbf{P} + \mathbf{x}$ and make it conform to the definitions of joints
Calculate $\tilde{\mathbf{y}}$
$\|\tilde{\mathbf{y}}\| < \varepsilon$?
No
Yes
End
Robot Description
$\xi^n, \xi^n_{st}, \mathbf{q}^n$
Measured Pose Data
Calculate g_i according to ξ_i
Identification
$\mathbf{x} = \left(\tilde{\mathbf{J}}^T\tilde{\mathbf{J}}\right)^{-1}\tilde{\mathbf{J}}^T\tilde{\mathbf{y}}$
Update
$q_i = q_i + \Delta q_i$
$\xi_{st} = \xi_{st} + \Delta\xi_{st}$
$\xi_i = Ad(g_i)\xi_i'(\Delta\boldsymbol{\eta}_i)$
$\|\mathbf{x}\| < \varepsilon$?
No
Yes
Terminate

Figure 2.1: Comparison between the least-square-based algorithms employed to solve the identification problem. On the left, the one proposed by He et al. (2010) uses both position and orientation measurements of the end-effector, and on the right, Wu et al. (2015) relies solely on the position of the end-effector.

- Correction

  The last phase requires implementing the corrected values into the arm's forward kinematics, mirroring the process employed with the DH convention.

Table 2.1: Comparison of open-loop kinematic calibration techniques partially retrieved from Kana et al. (2022). FK stands for forward kinematics.

Author	Measurement Technique	Calibration Method	Type of Regression
Kana et al. (2022)	Feedback from the robot controller	Compensating for the joint and link length offsets	Least-squares
G. Li et al. (2019)	Leica Geosystems Absolute Tracker (AT960)	Dual quaternion-based Forward Kinematics (FK) and calibration algorithm	Least-squares
Wang et al. (2014)	FARO arm to measure ball target position	POE and adjoint transformations based FK	Least-squares
C. Li et al. (2016)	FARO Laser Tracker ION	POE FK. Algorithm based on the ACS (axis configuration space) and adjoint error mode	Least-squares
Liu et al. (2018)	Leica Laser Tracker	POE FK	Least-squares
Balanji et al. (2022)	Monocular camera with ArUco markers	POE for FK and Perspective-n-Point (PnP) algorithm for pose estimation	Least-squares
Wu et al. (2015)	Romer Infinite 2.0 Measurement Arm	POE for FK, only position measurements	Least-squares
Wan et al. (2018)	Laser Tracker	Fiducial Localization Error (FLE) and closed-form iteration combined weighting algorithm	Gaussian process regression (GPR)-based volumetric error prediction

Gharaaty et al. (2018)	Creaform C-CTrack 780	Dynamic pose correction with PID controller	Root Mean Square (RMS) error minimization
Motta et al. (2016)	ITG ROMER	Levenberg–Marquardt algorithm to solve the regression problem	Non-Linear least-squares
Joubair and Bonev (2014)	Three-in datum spheres separated by precisely known distances measured on a CMM	Mathematical optimisation employing the condition number of the Jacobian J of the distances from the spheres	RMS error minimization
Lattanzi et al. (2020)	FARO Vantage Laser Tracker	Levenberg-Marquardt Mathematical optimisation	Non-linear least squares

2.3.3. Level 3: Non-Geometric Effects

Level 3 is the most complex degree of calibration, as it includes non-geometric factors, which are often non-linear. These effects include the elasticity of the joints and links, backlash in the motors' transmission units (Slamani et al., 2012), thermal expansion and slip-stick phenomena (Renders et al., 1991). Ziegert and Datseris (1988) firmly recommends that these non-linear error sources should be included in the model only if strictly necessary due to mainly two reasons: the increased level of complexity of the improved inverse kinematics, and the extra rotation parameters that would not be constant, like in the simpler calibration levels, but would instead depend on the other joint values.

- **Model**
  Kozlov and Klimchik (2022) states that for the elastic modelling of robotic arms, three methods exist: Virtual Joint Model (VJM), Matrix Structural Analysis (MSA), and Finite Element Analysis (FEA). FEA is the most precise and is usually employed at the latest design stage due to the high computational expenses required for high-order matrix inversion. VJM is the simplest method, as it is based on the extension of the traditional rigid model by adding localized springs to describe the elastic deformation of the links and the joints. This technique is widely used in serial robots and can hardly be applied to manipulators with more complex topology. Lastly, MSA is a compromise between the previous two methods, incorporating the main ideas of FEA with a larger number of elements, such as flexible links connected by the actuated and passive joints in the overall manipulator structure. Despite being commonly employed in advanced robotic systems, FEA and MSA won't be considered in this Thesis project since they necessitate the use of structural modelling and simulation software, which falls outside the project's scope.

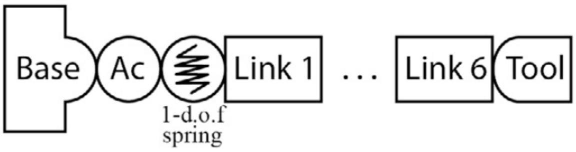


Figure 2.2: Schematic representation of the Virtual Joint Model of a robotic arm (Kozlov & Klimchik, 2022). Ac stands for ”Actuated Joints”.

As shown in figure 2.2, VJM consists of extending the rigid model of the robotic arm by adding a virtual spring after every joint, which will describe the elasticity of the relative joints/links.
The wrench $\mathbf{w}$ of the serial manipulator for the given configuration $\mathbf{q}$ can be expressed through the Cartesian Stiffness Matrix $\mathbf{K}_c$ and the end-effector deflection $\delta\mathbf{t}$ as follows:

$$\mathbf{w} = \mathbf{K}_c\delta\mathbf{t} \tag{2.35}$$

The elastostatic model for the i^{th} measurement can be therefore written as:

$$\delta\mathbf{t_i} = \mathbf{J}_{\theta,i}\mathbf{J}_{\theta,i}\mathbf{K}_\theta^{-1}\mathbf{J}_{\theta,i}^T\mathbf{w}_i \tag{2.36}$$

where $\delta\mathbf{t_i}$ is the end-effector displacement of the i^{th} measurement, $\mathbf{w}_i$ is the i^{th} wrench applied to the end-effector, and $\mathbf{J}_{\theta,i}$ is the Jacobian matrix for the configuration $\mathbf{q}_i$.
Equation 2.36 can be rewritten as:

$$\delta\mathbf{t_i} = \mathbf{A}_i c \tag{2.37}$$

where the vector $\mathbf{c}$ collects all the unknown compliance coefficients, and:

$$\mathbf{A_i} = [\mathbf{J}_{i,1}\mathbf{J}_{i,1}^T\mathbf{w}_i, \mathbf{J}_{i,2}\mathbf{J}_{i,2}^T\mathbf{w}_i, ..., \mathbf{J}_{i,1}\mathbf{J}_{i,1}^T\mathbf{w}_i] \tag{2.38}$$

Here $\mathbf{J}_{i,j}$ represents the columns of the Jacobian matrix $\mathbf{J}_{\theta,i} = [\mathbf{J}_{i,1}, \mathbf{J}_{i,2}, ..., \mathbf{J}_{i,n}]$.
The primary concern regarding the research conducted by Kozlov and Klimchik (2022) is its failure to account for the weight and resulting torque exerted on each joint due to the structure weight of the robotic arm itself. Nevertheless, two curious insights leading to a simplified model can be extracted from its work: firstly, there is no significant difference in considering only the deflections in the z-axis when compared to the x-y axes, and secondly, the final joint exhibits such minimal torque that excluding it from the computation paradoxically enhances the accuracy of the analysis.

A similar concept was followed by Caenen and Angue (1990), who integrated torques into the DH-formalism by adding torque-dependent offsets to the rotational DH parameters. Although they overlook torque equilibrium entirely, they do raise a valid observation: to fully characterize elastic deformations, a minimum of three additional parameters per joint must be determined.

Following their work, Khalil and Besnard (2002) implements the torque equilibrium to identify the correct DH parameters basing the model on lumped flexibility and assuming small deformations, modelling the revolute joints as linear torsional springs about the joint axis.
Assuming the external load as a lumped mass located at the endpoint of the robotic arm,

the mathematical model in their work first computes the forces and the moments of each link with respect to the base frame, then the deformation of both the links and the joints modelled as linearly elastic. However, this model does not account for the transversal deformations.

Kamali et al. (2016) follows a simpler approach, still modelling each joint i as a linear torsional spring with the following equation:

$$\tilde{\theta}_i = k_i \tau_i \tag{2.39}$$

where τ_i is the torque supported by the i^{th} joint.

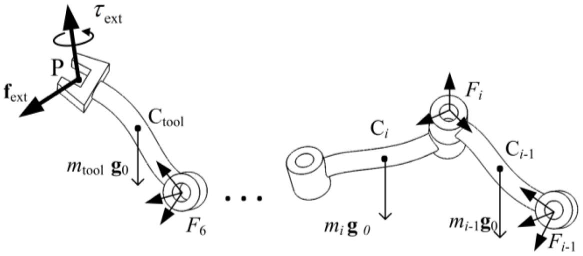

τ_{ext}
P
$\mathbf{f}_{\text{ext}}$
C_{tool}
$m_{\text{tool}}\ \mathbf{g}_0$
F_6
. . .
C_i
F_i
$\mathrm{C}_{i\text{-}1}$
$m_i\,\mathbf{g}_0$
$m_{i\text{-}1}\mathbf{g}_0$
$F_{i\text{-}1}$

Figure 2.3: Schematics of the serial robot arm links under external load and gravity effect, retrieved from Kamali et al. (2016).

The gravity vector with respect to frame F_i shown in Figure 2.3, can be obtained from:

$$\begin{bmatrix} \mathbf{g}_i \\ 0 \end{bmatrix} = (\mathbf{A}_i^0)^T \begin{bmatrix} \mathbf{g}_0 \\ 0 \end{bmatrix} \tag{2.40}$$

where $\mathbf{g}_0 = [0\ 0\ -9.81]^T$ is the gravitational acceleration vector, and $\mathbf{A}_i^0$ is the transformation matrix from the reference frame F_i to the base frame F_0.
The general $\mathbf{A}_j^i$ matrix is calculated from:

$$\mathbf{A}_j^i = \mathbf{A}_{i+1}^i \cdots \mathbf{A}_{j-1}^{j-2} \mathbf{A}_j^{j-1} \tag{2.41}$$

Defining $\mathbf{f}_j^i$ the gravitational force on link j with respect to frame F_i and $\mathbf{c}_j^i$ as the centre of gravity of link j with respect to F_i:

$$\mathbf{f}_j^i = m_j \mathbf{g}_i, \tag{2.42}$$

$$\begin{bmatrix} \mathbf{c}_j^i \\ 1 \end{bmatrix} = \mathbf{A}_j^i \begin{bmatrix} \mathbf{c}_j \\ 1 \end{bmatrix} \tag{2.43}$$

where m_j is the mass and $\mathbf{c}_j = [c_{x,j},\ c_{y,j},\ c_{z,j},]^T$ is the center of gravity of link j with respect to F_j. The torque generated by the gravitational force on link j on the axis of joint i can then be written as:

$$\tau_{i,ext} = [0\ 0\ 1]^T \cdot (\mathbf{p}^i \times \mathbf{f}_{ext}^i + \boldsymbol{\tau}_{ext}^i) \tag{2.44}$$

where:

$$\begin{bmatrix} \mathbf{f}_{ext}^i \\ 0 \end{bmatrix} = \mathbf{A}_{tool}^i \begin{bmatrix} \mathbf{f}_{ext}^{tool} \\ 0 \end{bmatrix} \tag{2.45}$$

$$\begin{bmatrix} \boldsymbol{\tau}_{ext}^{i} \\ 0 \end{bmatrix} = \mathbf{A}_{tool}^{i} \begin{bmatrix} \boldsymbol{\tau}_{ext}^{tool} \\ 0 \end{bmatrix} \tag{2.46}$$

$$\begin{bmatrix} \mathbf{p}^{i} \\ 0 \end{bmatrix} = \mathbf{A}_{tool}^{i} \begin{bmatrix} \mathbf{p}^{tool} \\ 0 \end{bmatrix} \tag{2.47}$$

Finally, the combination of all static torques caused by the external wrench and link weight supported by the i^{th} joint is calculated through the following equation:

$$\tau_i = \tau_{i,ext} + \sum_{j=i}^{N} \tau_{i,link\ j} \tag{2.48}$$

with N the number of links.
The estimated position of the end-effector can then be expressed as:

$$\mathbf{x}_{est} = f(\mathbf{v}, \mathbf{q}) \tag{2.49}$$

where v is the vector of all the robot parameters, and $\mathbf{q} = [\boldsymbol{\theta}^T\ \mathbf{W}^T]^T$ is the vector of joint angles and external wrench applied on the end-effector.

- **Measurement**
  As for the previous calibration level, it is necessary to record both the position and the orientation of the end-effector and/or other joints in different poses. However, often the non-geometric effects require more measurement devices specific to the effect that needs to be measured. To take into account elastic effects in complex systems, De Luca and Book (2008) suggests using semiconductor strain gauges (Klodmann et al., 2011) due to their high Gauge factors that lead them to produce a good signal-to-noise ratio, optical measurements of deflection, which are very versatile but are often subject to interference between the target measurement point and the sensor, or accelerometers like microelectromechanical systems (MEMS) that provide a low-frequency sensitivity and are very inexpensive, although they must be compensated for the gravity field, which requires priorly knowing the orientation of the sensor. Another approach is the one followed by Klodmann et al. (2011), who base their entire work on the torque feedback of the motors.

It is important to note that optimal measurement poses can be defined to properly identify the non-geometric coefficients. Joubair and Bonev (2013) defines five Observability Indices, based on the Singular Values Decomposition (SVD) of the identification Jacobian matrix $\mathbf{J}$ which represents the relationship between the pose errors in the calibration configurations and the parameters errors, and will be displayed in the following chapters. According to Joubair and Bonev (2013), five different observability indices can be computed for non-geometric parameter identification:

$$O_1 = \frac{\sqrt[m]{\sigma_1 \sigma_2 \cdots \sigma_m}}{\sqrt{(n)}} \tag{2.50}$$

$$O_2 = \frac{\sigma_m}{\sigma_1} \tag{2.51}$$

$$O_3 = \frac{\sqrt[m]{\sigma_1 \sigma_2 \cdots \sigma_m}}{\sqrt{(n)}} \tag{2.52}$$

$$O_4 = \frac{\sigma_m^2}{\sigma_1} \tag{2.53}$$

$$O_5 = \frac{1}{\frac{1}{\sigma_1} + \frac{1}{\sigma_2} \cdot \frac{1}{\sigma_m}} \tag{2.54}$$

with n the number of calibration measurements, m the number of parameters, and σ_i the singular values of the SVD:

$$\mathbf{J} = \mathbf{U\Sigma V}^T \tag{2.55}$$

with $\mathbf{U}$ and $\mathbf{V}$ orthonormal matrices, and Σ the following $n \times m$ matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_m \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{2.56}$$

- **Identification**
  Referring to the work of Kozlov and Klimchik (2022), Eq 2.37 leads to the following optimisation problem:

$$\hat{c} = (\sum_{i=1}^{n} \mathbf{A}_i^T \mathbf{A}_i)^{-1} (\sum_{i=1}^{n} \mathbf{A}_i^T \delta \mathbf{t}_i) \tag{2.57}$$

Khalil and Besnard (2002) identify for a 7 DOF robotic arm, specifically the PA-10, 46 independent calilbration parameters. Following a complex customized recursive Newton–Euler algorithm, similar to that used in the computation of the inverse dynamic models, the conclusion states that it is advisable to largely simplify the identification procedure by regarding some parameters as known rather than trying to identify all the parameters at once.

Kamali et al. (2016) followed the advice and narrowed the problem to 32 independent parameters by assuming the values of mass and centre of gravity of each link to be known, with only one per joint relative to the non-geometric elastic effect, and also using the DETMAX algorithm (Mitchell, 1974) to determine the 50 best configuration-wrench combinations out of a total of 875 possible combinations.
For the k^{th} measurement, the position error is noted as:

$$\Delta \mathbf{x}^k = \mathbf{x}_{measured}^k - \mathbf{x}_{est}^k \tag{2.58}$$

The goal is to identify the vector $\mathbf{v}$ of calibration parameters which leads to a null position error, which means solving the three (x,y,z) following non-linear equations:

$$\mathbf{x}_{measured}^k - f(\mathbf{v}, \mathbf{q}^k) = 0 \tag{2.59}$$

In order to simplify this problem, these equations are linearized around the nominal values as the initial estimation of the robot parameters. The differential form of Eq 2.49 for the k^{th} measurement can be expressed as:

$$\Delta \mathbf{x}^k = \mathbf{J}^k \Delta \mathbf{v} \tag{2.60}$$

where the Jacobian matrix $\mathbf{J}^k$ is obtained by differentiating 2.49 with respect to the calibration parameters as follows:

$$\mathbf{J}^k(\mathbf{v}, \mathbf{q}^k) = \frac{\partial f(\mathbf{v}, \mathbf{q})}{\partial \mathbf{v}}\Big|_{\mathbf{q}=\mathbf{q}^k} \tag{2.61}$$

For a set of n measurements, the following linear equation is obtained:

$$\Delta \mathbf{X} = \mathbf{J} \Delta \mathbf{v} \tag{2.62}$$

where the vector $\Delta \mathbf{X}$ includes all vectors of position errors of the end-effector:

$$\Delta \mathbf{X} = \left[(\Delta \mathbf{x}^1)^T \quad \Delta \mathbf{x}^2)^T \quad \ldots \quad \Delta \mathbf{x}^n)^T \right]^T \tag{2.63}$$

and $\mathbf{J}$ is the $3n \times m$ matrix:

$$\mathbf{J} = \left[(\mathbf{J}^1)^T \quad \mathbf{J}^2)^T \quad \ldots \quad \mathbf{J}^n)^T \right]^T \tag{2.64}$$

Eq 2.62 can be used iteratively to find the best estimation of the real robot parameters through the following procedure:

1. The initial value of $\mathbf{v}$ is set equal to the nominal values;
2. The identification Jacobian matrix $\mathbf{J}$ is calculated for the current $\mathbf{v}$ vector
3. The current $\mathbf{v}$ is used to find the estimated position of the end effector for all the selected configuration-wrench combinations (k = 1,...,n)
4. The vector of position errors $\Delta \mathbf{X}$ is calculated for all the measurements;
5. Similarly to what He et al. (2010) and Wu et al. (2015) have done as shown in Figure 2.1, the estimated value of $\Delta \mathbf{v}$ is calculated using the pseudo-inverse of $\mathbf{J}$ as follows:

$$\Delta \mathbf{v} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \Delta \mathbf{X} \tag{2.65}$$

6. $\mathbf{v}$ is updated adding the value of $\Delta \mathbf{v}$
7. Steps 2 to 6 are repeated until the norm of $\Delta \mathbf{X}$ converges to a stop criteria.

Tenhumberg and Bauml (2021) takes a different approach, formulating the following single combined least-squares problem based on all measurements and all markers to minimize the task space error, using the maximum a posteriori (MAP) approach:

$$\min_{\Theta} \left[\sum_{n}^{N} \frac{1}{\sigma_m^2} |y^{(n)} - h(q^{(n)}, \Theta)|^2 + (\Theta - \Theta_p)^T \Lambda_p^{-1} (\Theta - \Theta_p) \right] \tag{2.66}$$

where Θ are the optimal parameters, Θ_p is the mean of the prior Gaussian distribution, and $\Lambda_p = diag\ \sigma_p^2$ is the diagonal covariance matrix, with σ_p is the vector describing the uncertainty of the different calibration parameters.

It must be noted that because of the highly nonlinear measurement function, the use of multi-start for the initial guess is essential to guarantee a convergence to the global minimum.

- **Correction**

  Unlike calibrations of levels one and two, in this case the parameters cannot simply be added to the forward kinematics of the arm, as they strictly depend on the current joint arm configuration.

  Khalil and Besnard (2002) follows the iterative procedure shown in Figure 2.4. First, the forward kinematics assuming links and joints to be rigid is calculated, obtaining the position and orientation of each link; then, the deformations are computed and added to the rigid model; the new positions and orientations of each link are calculated, and compared to the ones obtained in the previous iteration; the iterations converge when the difference is below a certain threshold.

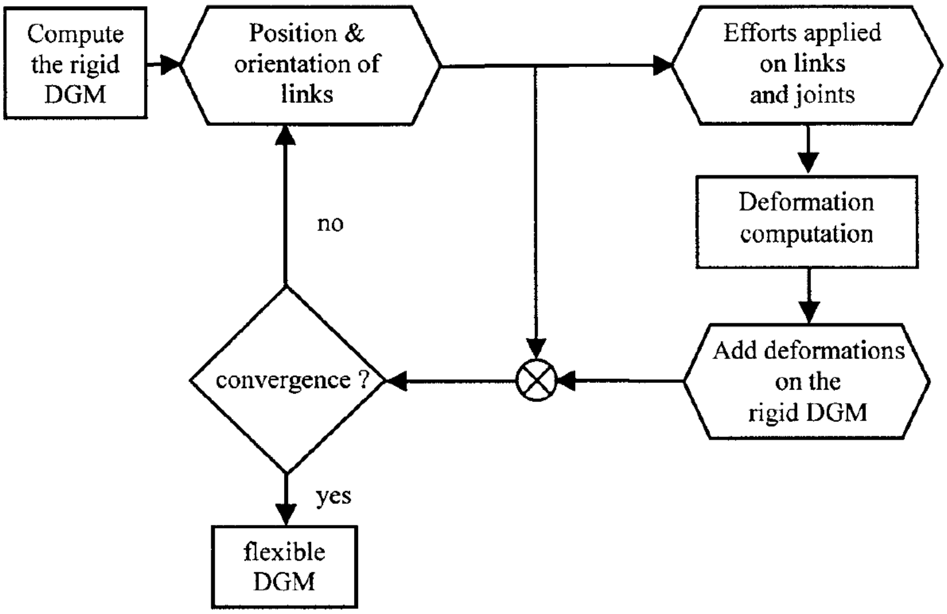

Compute the rigid DGM
Position & orientation of links
Efforts applied on links and joints
Deformation computation
Add deformations on the rigid DGM
convergence ?
no
yes
flexible DGM

Figure 2.4: Flowchart depicting the iterative procedure employed to rectify the forward kinematics of a robotic arm, addressing the elasticity present in both the joints and links. DGM stands for "Direct Geometric Model". Retrieved from Khalil and Besnard (2002).

Tenhumberg and Bauml (2021) follows a very similar approach, defining the DH parameters ρ as:

$$\rho = \rho(\rho_0, k, \tau(F(q,\rho), \nu)) \tag{2.67}$$

where k is the matrix containing the joints' elastic coefficients, and ν is the weight distribution. The equation must be solved iteratively as the DH parameters depend on the current torque τ acting on each joint. One solution is solving the following:

$$\rho_n = (1-\lambda)\rho_{n-1} + \lambda\rho(\rho_0, k, \tau(F(q,\rho_{n-1}), \nu)) \tag{2.68}$$

The iteration convergence is guaranteed by choosing the correct $\lambda \in [0,1]$, and using the nominal DH parameters as ρ_0.

2.4. Problem Statement

The primary objective of this work is to render the robotic arm showcased in Figure 1.5 fully operational and integrated into LRM. Although the arm will be described in more detail in the next chapters, it is necessary to note that the current inability to integrate the robotic arm with the rover stems from gravity-induced limitations. These limitations manifest as excessive bending of the 3D-printed links due to their own weight and elastic deflection within the joints from the inexpensive servo motors, especially when the arm is loaded. From the state of the art presented in this chapter, it is clear that the most direct and popular solution to such an issue would be to implement sensors in each joint and use the feedback of these sensors to develop a model-less compensation algorithm that would alter the nominal control loops of the arm.

Unfortunately, this relatively simple solution cannot be adopted in this particular scenario. LRM's strict cost requirement did not allow the integration of an expensive arm with complex motors equipped with a feedback system, and therefore, it employs 6 RC servo motors that lack such a feature. For this reason, the only possibility is a model-based calibration and gravity compensation, which will require the steps described in this chapter.

From these premises, the following research question and relative sub-questions arise:

How can a 6 DOFs 3D-printed robotic arm built with inexpensive motors improve its accuracy through a model-based joint deflection compensation?

Rationale: The LRM rover aims to utilize a 3D-printed robotic arm for manipulation tasks. However, its inherent flexibility caused by the printed material and the inexpensive motors used in its joints significantly hinders its accuracy. By focusing on this aspect, this research aims to bridge the gap between the cost-effectiveness of 3D-printed arms and the need for accurate manipulation, delivering a fully operational robotic arm. Due to the absence of feedback sensors on the servo, using a model-less compensation is not feasible. Therefore, a model-based approach is needed, with the consequent use of external measurement devices.

SUBQ-1 Considering the limitations of inexpensive RC servo motors, what model-based approach can effectively compensate for elastic deflection within the joints under varying load conditions?
Rationale: The state of the art in robotic arms calibration and gravity compensation suggests that model-based compensation is the best approach to this scenario. However, literature often concerns industrial robotic arms, which present linear and predictable behaviour compared to less expensive hardware, like the one implemented in the LRM manipulator. One of the goals of this thesis will be to understand how and to what extent existing model-based approaches can be adapted to this robotic arm, carefully choosing the most suitable parametrization.

SUBQ-2 To what extent should the model developed for joint deflection compensation be simplified, considering the trade-off between accuracy and computational complexity suitable for the intended control system of the robotic arm?
Rationale: Considering the limited accuracy of the tracking device that will be used for the ”Measurement” phase and the unpredictability of the anisotropic 3D-printed plastic the robotic arm is made of, without adequate consideration, the model may reach a level of complexity not sustainable for the time scope and the resources available for this project. Therefore, it is crucial to make reasonable assumptions upfront to simplify the problem.

SUBQ-3 How should the common issues of target occultation and collision avoidance between the arm and its surroundings during measurement be addressed?
Rationale: The measurement phase will require testing the arm in different configurations. Without a proper feasibility analysis, the risk of collisions or occlusion of the target by the arm or nearby structures becomes significant. These issues could lead to inaccurate measurements, system malfunctions, or even irreparable damage to the arm. To mitigate these risks, a comprehensive strategy should be employed, including path planning algorithms and a preventive collision detection analysis.

SUBQ-4 To what extent does joint elasticity influence the position of the end effector, compared to kinematic offsets caused by mechanical defects?
Rationale: Comparing different parameterizations allows for assessing whether correcting for joint elasticity significantly impacts the position of the end-effector, relative to the kinematic offsets caused by mechanical defects. This will help evaluate the necessity of compensating for elasticity.

SUBQ-5 What experimental validation methods can be used to assess the effectiveness of the gravity compensation strategy implemented, evaluating its impact on arm performance in various load scenarios?
Rationale: The last step would be validating the arm's increased capability by comparing its performance enabling and disabling the gravity compensation algorithm. Different sample targets will be needed to ensure the algorithm works well regardless of the load scenario.

SUBQ-6 How can the proposed model-based calibration and gravity compensation be effectively integrated into the control system of the robotic arm?
Rationale: Following the development of the flexibility compensation algorithm, the inverse kinematics of the arm and the newly developed algorithm will be integrated into the rover control system.

2.5. Research Methodology and Chapter Organisation

This section outlines the approach that will be adopted to achieve the research objectives as well as the organisation of this report.

The research begins in this chapter with a comprehensive review of the current state-of-the-art in robotic arm calibration and compensation techniques, especially methods for correcting elastic and geometric deformations. The review examines the relevant literature on forward and inverse kinematics, emphasising the DH convention and the POE method, providing the mathematical background necessary to understand the calibration and compensation processes used in this work.

Following the literature review, Chapter 3 involves the 1st level calibration of the LRM robotic arm, focussing on optimising the robotic arm's basic capabilities and ensuring proper operation before tackling complex compensation. The first step of the calibration involves analyzing the Pulse With Modulation (PWM) signals driving the RC servos, ensuring that the correct signals correspond to the desired joint angles. This initial calibration links each PWM signal to a specific angle for the servos, allowing the robotic arm to operate with basic accuracy and functionality. Although this step does not refer to any particular subquestion, it is fundamental to ensure the success of the more advanced calibration levels.

Once the basic calibration is completed, Chapter 4 transitions into developing the elasto-kinematic model of the robotic arm, addressing **SUBQ-1**. The goal is to build a kinematic model that captures both the mechanical behaviour and the elastic distortions, balancing accuracy with computational complexity to ensure the model is feasible for real-time implementation.

Once the model is established, Chapter 5 shifts to data collection and measurement. Following some preliminary tests to determine the functionality of the advanced real-time tracking system, as well as the linearity of the elasticity of the RC servos, answering the **SUBQ-2**, the exact position and orientation of the Tool Center Point (TCP) are measured in multiple predefined poses of the robotic arm. These measurements, which required an analysis of the arm's workspace addressing **SUBQ-3**, will help in identifying errors between the theoretical kinematic model and the actual performance of the arm, which are then processed to extract the parameters necessary for building the error correction model.

Next, in Chapter 6, the parameters are identified through a least squares optimisation using part of the measured poses, and using the rest to validate the arm's performance under various load conditions, and using different sets of parameters. This methodology addresses both subquestions **SUBQ-4** and **SUBQ-5**, enabling effective validation and comparison of the impacts of elastic deformations versus kinematic offsets.

After validating the compensation model, Chapter 7 deals with its integration into the LRM software architecture, adding the feature of inverse kinematics to the Links and Nodes Manager, answering the last subquestion **SUBQ-6**. The control system is updated to include real-time corrections, allowing the robotic arm to adjust its TCP dynamically. The integration process also involves testing the system's ability to avoid collisions and manage complex movements, ensuring that the calibrated arm functions effectively in real-world applications.

Finally, the research concludes by summarizing the key outcomes and providing recommendations for future work. Potential improvements include refining the compensation model, integrating additional sensors, or exploring more advanced calibration techniques to further enhance the robotic arm's accuracy and robustness.

3

LRM Arm and 1^{st} Level Calibration

This chapter will briefly describe the capability optimisation conducted before the calibration process and the first-level calibration of the RC servos implemented in the robotic arm's joints. It will introduce the communication protocol adopted to move the motors into the desired position, highlighting the primary sources of inaccuracies. This process involves positioning the servos to known reference points and fine-tuning their response to ensure precise movement and alignment. The primary goal of this calibration step is to minimize discrepancies between the commanded positions and the actual positions of the servos. By establishing accurate reference points and adjusting the servo behavior accordingly, a solid foundation is laid for the subsequent calibration levels.

3.1. Capability optimisation

Before calibrating the arm, a preliminary step was taken: updating the link lengths to optimise its capability and reachability. This critical step was necessary to avoid a second calibration in the near future and to acquire real data on the arm's capability, which will be used for the validation phase.

Although the task was approached in two different ways, both methods focused on the same objective and optimisation targets. The objective was to optimise the orientation of the first link relative to the rover's main body, along with the lengths of the second and fourth links. The optimisation targets were the critical areas the arm needed to reach: the ground behind the rover and the basket on the rover's back, where the sample is placed after retrieval.

The first approach involved testing 10 million random poses to observe how frequently the end effector landed in the target areas. These results were then compared across different arm configurations, explicitly testing the orientation of the first link at 45 and 60 degrees, and adjusting the lengths of the second and fourth links by 1 cm and 2 cm. Longer links would lead to stability issues during movements. The results were then plotted, and an example is shown in Figure 3.1 and 3.2.

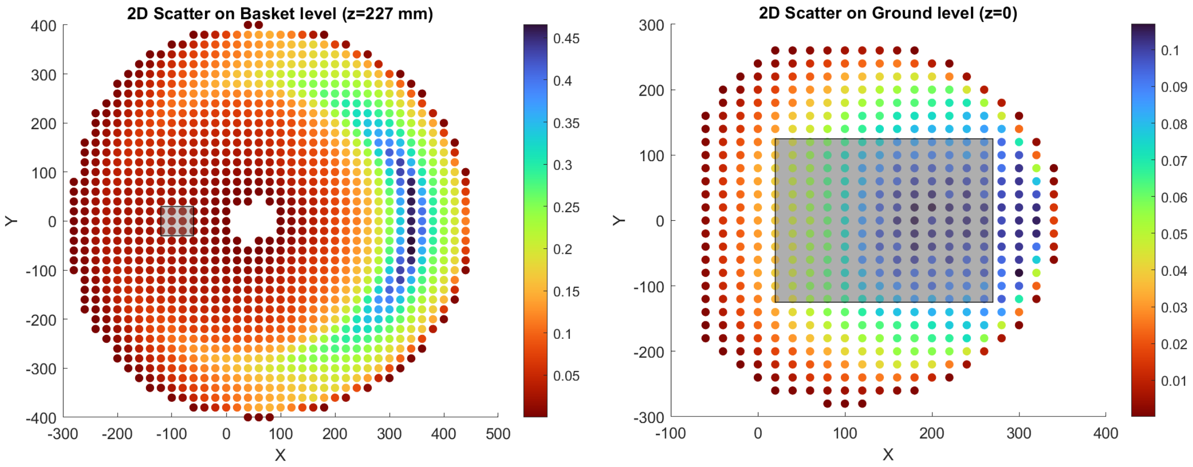


Figure 3.1: 2D Scatter Plots of Robotic Arm Reachability. The plot represents a meshed area of the workspace, where the colour of each dot indicates the frequency of arm reachability within that region. The grey area marks the position of the collection basket on the rover on the left, and the target ground zone on the right.

The random sampling approach tends to bias the results towards configurations closer to the arm's home position or default setup. This happens because random poses are more likely to fall within areas that require minimal movement from the home position, leading to a concentration of successful end-effector positions in that vicinity. Consequently, instead of producing plots where the "blue" area representing reachability is a well-distributed and circular shape, the results show a dense cluster of successful positions around the home configuration. This clustering effect skews the analysis, making it appear that the arm is more capable in certain areas than it is while potentially underestimating its reach in more distant or complex regions. As a result, this method does not effectively capture the full range of the arm's reachability or its ability to operate in critical target areas.

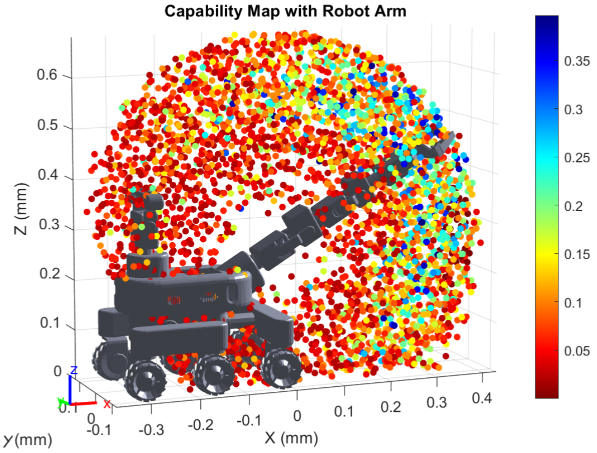


Figure 3.2: 3D Scatter Plot of the Robotic Arm Reachability. Like Figure 3.1, it represents a meshed area of the workspace, where the colour of each dot indicates the frequency of arm reachability within that region.

The second approach took a more systematic and targeted method. Instead of relying on random poses, this approach used inverse kinematics to analyze a grid of points within the critical areas: the basket and the ground behind the rover. For each point on the grid, six different orientations of the end effector were tested. If the inverse kinematics algorithm could find a valid solution for the arm to reach that specific point and orientation, the point on the grid would be marked with a +1 value. This process was repeated across the entire grid, accumulating values that reflected the arm's ability to successfully reach and orient itself at various locations within the target areas. After completing this grid-based analysis for each arm configuration, the results were summed to give a total score representing the arm's overall reachability and effectiveness for that specific setup. By comparing these scores across different configurations, adjusting the orientation of the first link and the lengths of the second and

fourth links, the configuration with the highest total score was identified as the best trade-off between reachability and capability.
The final result was to adjust the orientation of the first joint from the original 45 degrees to 60 degrees, and to increase the lengths of the second and fourth links by 1 cm and 2 cm, respectively.

3.2. RC servos

RC servos are essential in various modern electronics and robotics applications, providing precise control over movement and positioning. These small yet powerful devices are integral to everything from hobbyist model aeroplanes and cars to advanced robotics and automation systems. An RC servo, a precision-driven rotary actuator, comprises three key elements: a DC motor, a set of gears, and a feedback control system. The servo motor powers a set of gears that convert the motor's rotation into the exact angular movement of the servo arm. The feedback control system, typically a potentiometer, plays a pivotal role in this process, constantly monitoring the servo's position and adjusting the motor's operation to maintain the desired position.

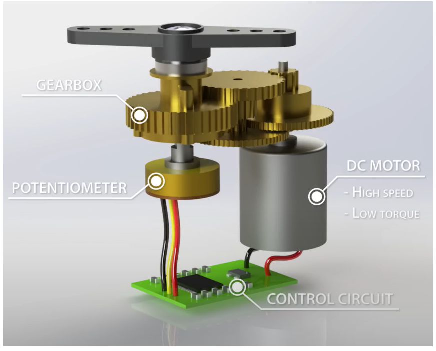


Figure 3.3: CAD model of the internal components of an RC servo. Retrieved from https://howtomechatronics.com

The core functionality of an RC servo is based on Pulse Width Modulation (PWM). The servo's position is controlled by varying the width of the PWM signal sent to it. The servo generally interprets the pulse length to determine the angle at which it should position its output shaft.
RC servos are highly valued for their accuracy and reliability. They are commonly used in radio-controlled vehicles, controlling various functions such as steering, throttle, and other movable parts. In robotics, servos enable precise control of limbs and joints, making them ideal for applications requiring accurate and repeatable movements.

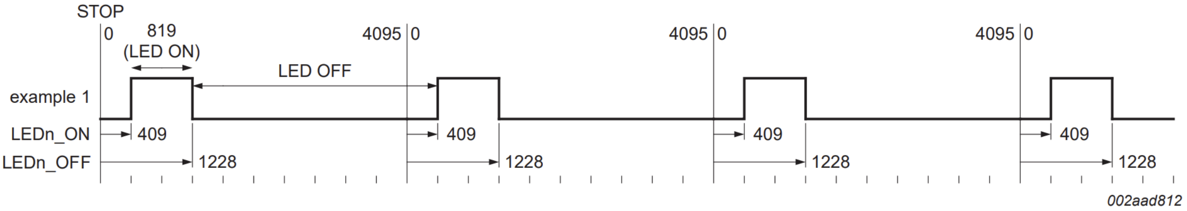


Figure 3.4: Visualization of the PWM signal as a function of the LED_ON and LED_OFF values written into the port expander registers.

3.3. LRM servo control

Due to the high torque requirements of the LRM's arm, specialized servos were needed. The AGF High Torque Series (AGFRC, 2024) was chosen for this purpose, as it also offers the capability to implement a soft start feature, which helps manage the initial load and ensures smoother operation. Adding the gripper and pan-tilt servos, the LRM must control a total of nine motors simultaneously. This is accomplished using the PCA9685 port expander, which interfaces with the servos via an I²C bus. The PCA9685 generates distinct PWM signals for each servo by outputting a separate signal on each connected data pin.

One of the primary sources of inaccuracy originates from the port expander, specifically in converting the desired PWM signal into the actual ”off-time” of the square wave. To achieve accurate PWM generation, the frequency of the PWM signal must first be set by adjusting the chip’s clock speed through a prescale value. This process is governed by the following formula:

$$prescale = \frac{CLOCK\_SPEED}{(4096 * freq * 0.948) - 1} \tag{3.1}$$

where $freq$ is the I²C frequency. The off-time value is the function of the real off-time and the frequency deriving from applying the previously calculated prescaler, using:

$$OFF\_TIME = \frac{4096 * freq * \text{PWM}}{1000000} \tag{3.2}$$

However, inaccuracies arise due to two primary factors. First, the prescaler must be an 8-bit integer, which necessitates rounding and can lead to errors if the calculated prescaler value is not a whole number. Second, the actual clock speed of the PCA9685 is rarely precisely 25 MHz, as it fluctuates around this nominal value. These variations contribute to inaccuracies in the PWM signal, affecting the precise positioning of the servos and leading to the need for a 1st level calibration.

3.4. RC Servos Calibration

To conduct this calibration, the device shown in figure 3.5 has been designed. It consists of a 3D printed protractor with a pointer screwed into the servo head. The sensitivity of 2 degrees allows for a precise first-level calibration by manually identifying which PWM signals match some pre-defined waypoints. To cover the vastest range, the chosen angles were the mid position, 0 degrees, and the servo limits -100 and 100 degrees.

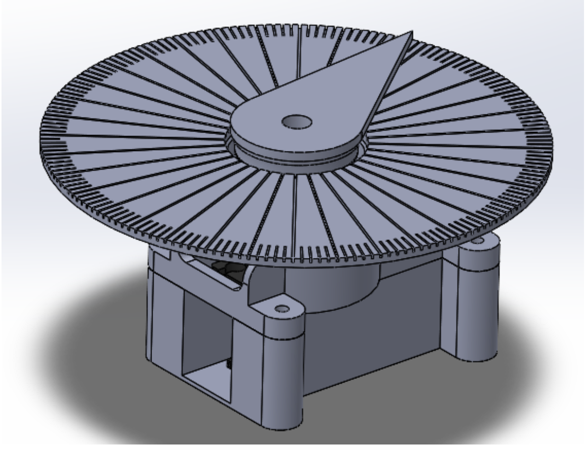


Figure 3.5: The 3D printed caliber-like calibration device used to compare the PWM period sent by the main PCB of LRM, with the real angle of each servo.

Angle	-100° $[\mu s]$	0° $[\mu s]$	100° $[\mu s]$
Joint 1	620	1540	2460
Joint 2	620	1550	2450
Joint 3	630	1555	2455
Joint 4	600	1505	2425
Joint 5	560	1490	2380
Joint 6	565	1495	2395
TCP	580	1480	2380

Table 3.1: Results of the 1st level calibration. Each joint needs a different PWM signal for the -100,0 and 100 degrees positions.

Although servo manufacturers standardize the PWM signal’s off-time for the mid position to correspond to 1500 μs, the imprecision of the servos was evident from the first use of the calibration device. Not only did the mid position not match the standard one, but each servo had a slightly different behaviour. The results of the manual calibration are presented in Table 3.1.

Once these reference points have been identified, the arm’s control software linearly interpolates the data with different slope values for negative or positive angles.

4

The Elasto-Kinematic Model

This chapter will delve into the more advanced calibration levels, which will take into account not only the angular and linear offsets resulting from inaccuracies in the assembly and manufacturing processes but also the error in the position of the end effector deriving from the intrinsic elasticity of the joints and the links. The chapter consists of the first calibration step described in Chapter 2: Modelling the elasto-kinematic model and parametrising the errors.

4.1. Kinematic Model

The necessary model consists of not only the forward kinematics model, which describes the position of the end-effector as a function of the joint angles, but also the modelling of the possible errors that can affect the precision of the forward kinematics. Without a proper parametrization of the errors, the calibration process would not converge, leading to incorrect results. It is the case of Khalil and Besnard (2002), who identified negative values for the elastic coefficients of the joints for a PA-10 robotic arm.

This section consists of three subsections: the arm's kinematic modelling, an explanation of the linear and angular offsets corrected by the second-level calibration, and a description of the elastic model compensated by the third-level calibration.

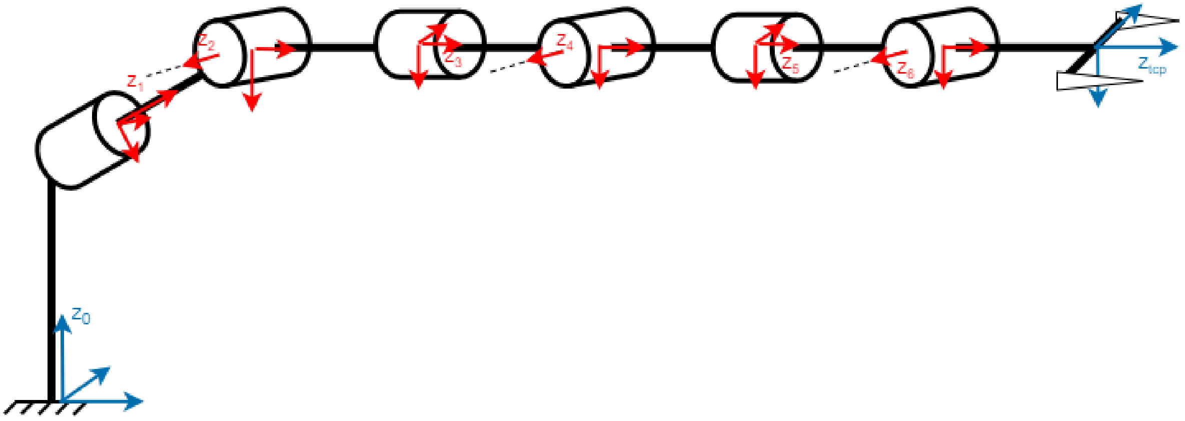


Figure 4.1: Schematic representation of the kinematic model of LRM robotic arm, showing the 6 degrees of freedom and the orientation of each joint.

As carefully described in Chapter 2, although the POE avoids the problem of singularities, the complexity of its formulation when compared to DH, as well as the fact that the control software of the manipulators used in the Robotics Institute is based on DH descriptions, ultimately led to the choice of DH over POE. However, the conventional DH parameterization cannot accurately describe the positions of joints 3 and 5, as it collapses them onto the previous joints (2 and 4). This limitation arises because the DH convention does not account for translations along the y-axis of the preceding joints. To avoid this issue, extra translation parameters are necessary. The same concept applies to the translation of the base and the end effector, adding a rotation parameter for each axis. Considering the added parameters the kinematic description takes the name of Modified Denavit Hartenberg (MDH).

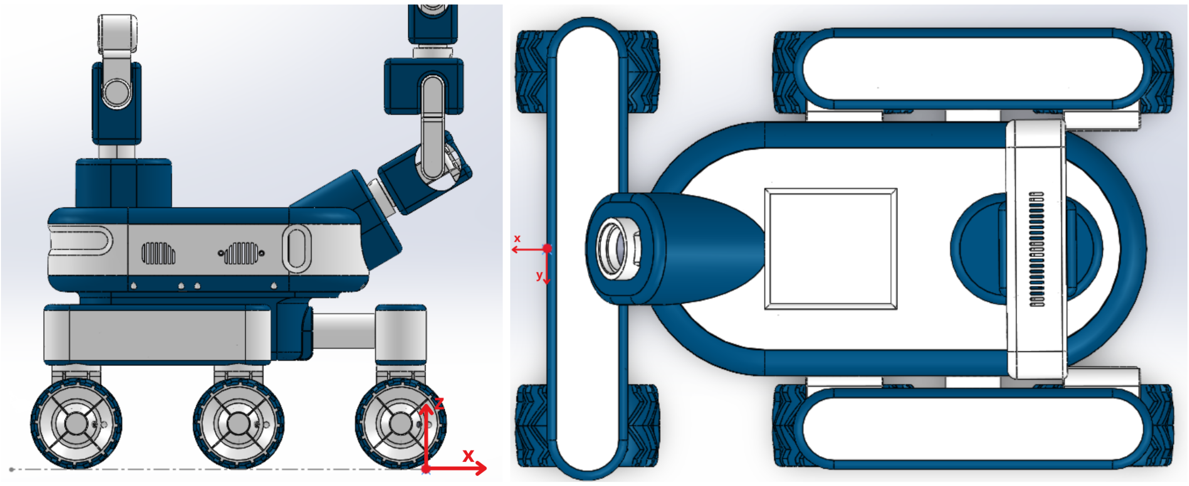


Figure 4.2: Render of the CAD model of LRM, showing the position at the ground level of the base reference frame, highlighted in red, used for the forward kinematics of the arm.

Joint	α	**a [mm]**	θ	**d [mm]**
1	$\pi/2$	0	$\theta_1 + \pi$	46.905
2	$-\pi/2$	0	θ_2	0
3	$\pi/2$	0	θ_3	43.069
4	$-\pi/2$	0	θ_4	0
5	$\pi/2$	0	θ_5	43.250
6	0	0	θ_6	0

Table 4.1: Table containing the DH parameters used to describe LRM Robotic Arm.

-	α	β	γ	x [mm]	y [mm]	z [mm]
Base	0	$\pi/3$	0	-34.975	0	236.08
TCP	$-\pi/2$	0	0	0	0	130.38
Joint 3	0	0	0	0	0	103.51
Joint 5	0	0	0	0	0	107.75

Table 4.2: Table with the extra DH parameters which describe the position of the base, the TCP, and the third and fifth joints. Due to these parameters, the new model takes the name of modified DH (MDH).

4.2. 2nd Level: Kinematic Offsets

The second level calibration aims to compensate for the offsets derived from the manufacturing and assembly procedures. Advancements in 3D printing techniques have achieved a precision of tenths of millimetres, such as the machine used for LRM's arm, which prints in ASA. However, the stacking of hot filament needed to create the elongated components used in the arm and the support material needed to guarantee stability during the process have side effects that affect the printing quality. For example, the coarse surface hidden by the support material inevitably hinders the precision of two components screwed together by those surfaces. The issue may be solved by manually sending the parts, but there is the risk that they will not fit perfectly any longer. The same applies to the numerous plastic parts in which metal components fit precisely without backlash. Although the plastic will easily bend to welcome the bearing, or any other metal insert, this deformation may affect the positioning of any plastic component mounted on it.

Also, during the assembly process, the scarce rigidity of ASA can easily result in misalignments and imprecisions. Over-tightening a screw, pressing a bearing with too much force, or placing a cable in the wrong spot, leads to undesired offsets in the kinematic chain and to errors in the position of the end effector.

To tackle this, the calibration procedure needs a model of this phenomenon. The most direct solution would be adding six offset parameters for each joint, representing the error in the x,y,z direction and orientation. However, the DH parametrization offers the possibility of using its layout instead of adding three extra translations and rotations per joint. Table 4.3 shows all of the kinematic offset parameters, for a total of 38, which must be added to the ones presented in table 4.1 to obtain the corrected MDH parametrization.

4.3. 3rd Level: Elasticity

As anticipated in the previous chapters, the initial qualitative tests of LRM's arm clearly showed how the load on the TCP heavily influenced the displacement of the end effector in different poses.

The first step in considering these deformations is modelling them. Industrial robotic arms are usually made of metal components and torque feedback sensors, which heavily dampen the effect of the material's elasticity. Instead, this peculiar and extreme case scenario, consisting of a plastic arm with inexpensive joint motors, requires a specific model as it is hardly represented

Joint	$\boldsymbol{\alpha}$	a [mm]	$\boldsymbol{\theta}$	d [mm]
1	$\delta\alpha_1$	δa_1	$\delta\theta_1$	δd_1
2	$\delta\alpha_2$	δa_2	$\delta\theta_2$	δd_2
3	$\delta\alpha_3$	δa_3	$\delta\theta_3$	δd_3
4	$\delta\alpha_4$	δa_4	$\delta\theta_4$	δd_4
5	$\delta\alpha_5$	δa_5	$\delta\theta_5$	δd_5
6	$\delta\alpha_6$	δa_6	$\delta\theta_6$	δd_6

-	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	x [mm]	y [mm]	z [mm]
Base	$\delta\alpha_b$	$\delta\beta_b$	$\delta\gamma_b$	δx_b	δy_b	δz_b
TCP	$\delta\alpha_t$	$\delta\beta_t$	$\delta\gamma_t$	δx_t	δy_t	δz_t
Joint 3	-	-	-	-	-	δdd_3
Joint 5	-	-	-	-	-	δdd_5

Table 4.3: Summary of all the 38 parameters derived from the parametrization for second-level calibration.

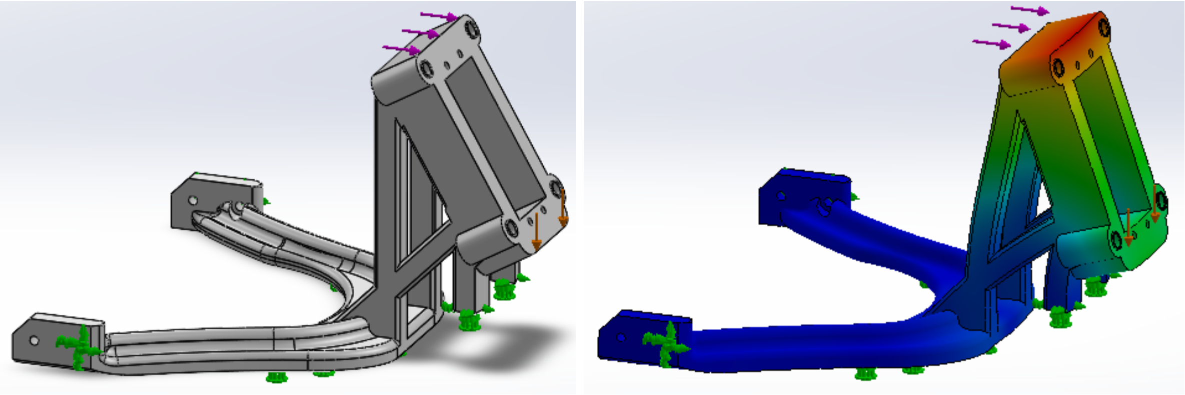


Figure 4.3: Qualitative FEA simulation of the elastic deformation of the base structure holding the arm, when under load, showing its deformation that must be parametrized in the elastic model.

in the literature. Both the joints and the links are expected to flex, however, how much one will bend with respect to the other is unknown and unpredictable. The safest and most reliable

approach is to model only the rotational elasticity of each joint, while neglecting the flexibility of the links in the other two directions.
This means that the only difference from the parameters shown in Table 4.3 is the addition of the following six parameters $\mathbf{k}$:

- $\mathbf{k_1}$: $\delta\theta_1 + k_1 \cdot \tau_1$
- $\mathbf{k_2}$: $\delta\theta_2 + k_2 \cdot \tau_2$
- $\mathbf{k_3}$: $\delta\theta_3 + k_3 \cdot \tau_3$
- $\mathbf{k_4}$: $\delta\theta_4 + k_4 \cdot \tau_4$
- $\mathbf{k_5}$: $\delta\theta_5 + k_5 \cdot \tau_5$
- $\mathbf{k_6}$: $\delta\theta_6 + k_6 \cdot \tau_6$

On top of this, to consider the lateral deformations of the plastic components, and as a result of the first qualitative observations, it was decided to add one more elastic coefficient to the base structure holding the weight of the entire arm:

- $\mathbf{k_0}$: $\delta\beta_b + k_0 \cdot \tau_0$

bringing the total number to 45 parameters. The effect of the torque around the y-axis on this component is shown in Figure 4.3. As described in Chapter 2, the basic kinematic model must be upgraded to consider the joints' elastic coefficients and the base structure. To do so, it is possible to use the rigid model as the initial configuration, calculating the torques acting on each joint and the expected deformations. The forward kinematics are updated with the new deformation, and the process is repeated until convergence.

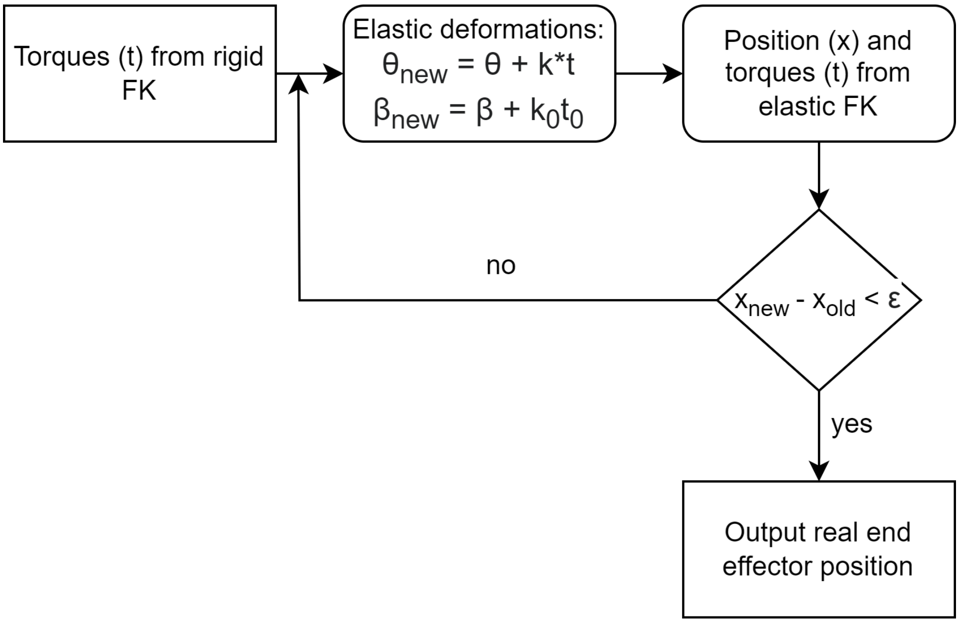

Torques (t) from rigid FK
Elastic deformations:
θ$_{new}$ = θ + k*t
β$_{new}$ = β + k$_0$t$_0$
Position (x) and torques (t) from elastic FK
x$_{new}$ - x$_{old}$ < ε
no
yes
Output real end effector position

Figure 4.4: Flowchart of the elastic correction in the forward kinematics of the arm. ε is an arbitrary small number, set to 10^{-3} mm.

4.4. Center of Mass Identification

The last step in completing the kinematic model of the arm is to identify the real position of the centre of mass of each joint. Considering the complex shapes and the number of components adding up to the weight of each link, obtaining it analytically is impossible. The remaining solutions are two.

- **CAD method**: This approach utilizes a high-fidelity CAD model of the link that includes all elements such as screws, inserts, cables, bearings, and, crucially, the precise 3D printing characteristics of each ASA part. By incorporating these detailed components into the CAD model, the centre of mass can be calculated with a high degree of accuracy. However, this method demands comprehensive knowledge about the centre of mass for each individual component, information that is often not readily available in standard datasheets.

- **Spin Method**: This empirical technique involves rotating the link around different axes until it ceases to oscillate and achieves a steady, free rotation. By performing this procedure around two orthogonal axes, the centre of mass can be pinpointed at their intersection. This method is advantageous because it does not require detailed component-level data and relies on direct physical observation and adjustment.

Given the challenges associated with the CAD method, primarily the difficulty in obtaining the precise centre of mass data for each component, the spin method was selected as the preferred solution.
Each link has been set up with local reference frames and a certain amount of pre-defined holes where the tips of a turning machine will be skewered. During each test, the rotation is carefully monitored. When the link rotates smoothly and consistently without any oscillation, it indicates that the centre of mass has been effectively aligned with the axis of rotation. Once the two axes were identified, the position of the centre of mass with respect to the previous joint was extracted from the CAD. The centre of mass of the end effector is calculated as the weighted average of its structure and the load attached to it.

4.5. Parameter Dependencies

Once the kinematic model is complete, a crucial step is to understand which of the 45 parameters shown in Table 4.3 are linearly dependent. This is necessary to ensure that the identification process runs smoothly and does not end in problems of matrix ill-conditioning. To do so, a valid instrument is the Jacobian Identification Matrix introduced in Chapter 2 as the link between the perturbation in the value of the parameters and the inaccuracy in the position of the end-effector of the robotic arm:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathrm{FK}(\mathbf{q_1},\boldsymbol{\nu})}{\partial \nu_1} & \frac{\partial \mathrm{FK}(\mathbf{q_1},\boldsymbol{\nu})}{\partial \nu_2} & \cdots & \frac{\partial \mathrm{FK}(\mathbf{q_1},\boldsymbol{\nu})}{\partial \nu_{45}} \\ \frac{\partial \mathrm{FK}(\mathbf{q_2},\boldsymbol{\nu})}{\partial \nu_1} & \frac{\partial \mathrm{FK}(\mathbf{q_2},\boldsymbol{\nu})}{\partial \nu_2} & \cdots & \frac{\partial \mathrm{FK}(\mathbf{q_2},\boldsymbol{\nu})}{\partial \nu_{45}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{FK}(\mathbf{q_K},\boldsymbol{\nu})}{\partial \nu_1} & \frac{\partial \mathrm{FK}(\mathbf{q_K},\boldsymbol{\nu})}{\partial \nu_2} & \cdots & \frac{\partial \mathrm{FK}(\mathbf{q_K},\boldsymbol{\nu})}{\partial \nu_{45}} \end{bmatrix} \tag{4.1}$$

where $\boldsymbol{\nu}$ is the vector containing all the calibration parameters, and $\frac{\partial \mathrm{FK}(\mathbf{q},\boldsymbol{\nu})}{\partial \nu_i}$ is a $[3 \times 1]$ column

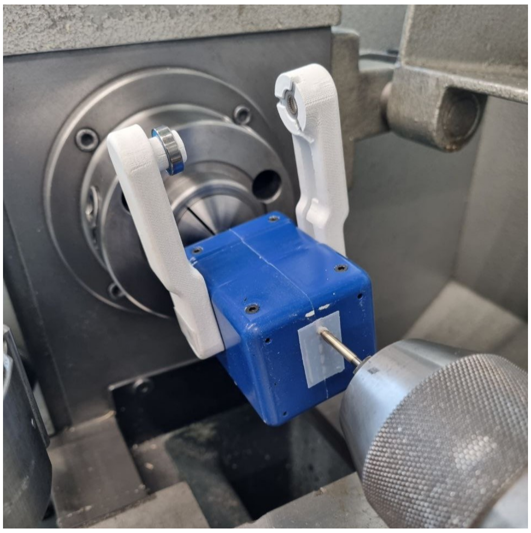


Figure 4.5: Technique used to measure the position of the centre of mass of each link. The component is left hanging between the two metal tips attached to a turning machine; when it stops oscillating, one axis of inertia is defined. When two or three are defined, it is possible to pinpoint the position of the centre of mass.

Link	x [mm]	y [mm]	z [mm]	mass [g]
1	0	2.0	37.3	123
2	12.5	-0.2	47.6	177
3	0	2.5	31.9	102
4	4.5	0	53.1	133
5	0	0	29.1	106
6	0	0	45.1	81

Table 4.4: Table showing the position of the center of mass for each link relative to the reference frame of its corresponding joint, along with the respective masses.

vector of the end effector position partial derivatives for the k^{th} pose **q**. If the calibration process were to consider also the orientation of the end effector, then $\frac{\partial \text{FK}(\mathbf{q},\boldsymbol{\nu})}{\partial \nu_i}$ would be a $[6 \times 1]$ column vector. These derivatives must be calculated using numerical approximations, specifically finite differences. The central finite difference was chosen to ensure the most reliable approximation using the following formula:

$$\frac{\partial \text{FK}(\mathbf{q},\boldsymbol{\nu})}{\partial \nu_i} = \frac{\text{FK}(\mathbf{q},\nu_i+\varepsilon)-\text{FK}(\mathbf{q},\nu_i-\varepsilon)}{2\varepsilon} \tag{4.2}$$

The variable ε is the ”small enough” perturbation factor, or step size, used in the finite difference calculations. Although this topic is severely neglected in the literature regarding the calibration of robotic arms, it was immediately clear it needed a deeper analysis, which is presented in the following subsection 4.6. Once the Identification Matrix is obtained, studying its rank makes it possible to understand which columns are linearly dependent and which parameters can be removed from the analysis. Although theoretically, keeping those parameters does not influence the end result, as it would simply mean assigning random but coupled values to the linearly dependent parameters, it represents a serious threat to the main step of the identification process, the computation of the inverse of the identification matrix. A matrix with linearly dependent parameters is referred to as singular. It would be impossible to compute the inverse of such a matrix using numerical methods, and even if the matrix is nearly singular, it would require dividing by values very close to zero, leading to numerical stability issues.

One solution to identify and eliminate the linearly dependent columns is removing each parameter one by one and calculating the rank of the matrix. This process is iterated until the rank matches the number of columns.

However, as previously mentioned, this procedure does not exclude the parameters that cause the determinant to be too close to zero. In order to tackle this issue, the singular values of the SVD of the matrix can be analysed. Again each parameter is removed one by one until all the singular values lie above a certain threshold. To guarantee numerical stability, this threshold

was empirically found to be 0.95.
Another workaround to solve this issue is to use the built-in tolerance factors that Matlab allows when calculating the pseudo inverse of non-squared matrices. This tolerance basically ignores the singular values below the threshold, allowing for the pseudo inverse to be computed at the expense of not knowing which parameters cause the matrix's ill-conditioning.
The first approach was preferred, so fewer parameters were needed in the following algorithms, which heavily lightened the computational costs.
The result is an astonishing reduction from 45 parameters to 28.

$\delta\alpha_2$	$\delta\alpha_3$	$\delta\alpha_4$	$\delta\alpha_5$	k_0	k_1	k_2	k_3	k_4	k_5	k_6	δa_3	δa_4	δa_5
$\delta\theta_1$	$\delta\theta_2$	$\delta\theta_3$	$\delta\theta_4$	$\delta\theta_5$	$\delta\theta_6$	δd_4	δd_5	δx_b	δy_b	δz_b	δy_t	δz_t	δdd_3

Table 4.5: List of the strongly independent parameters which lead to singular values of the Identification Matrix SVD above 0.95.

It must be noted that these parameters will have the same unit of measurement of the relative dimension to be corrected in the kinematics. Therefore, the linear offsets will be in mm, while the angular ones will be in radiants. This means that the angular elastic coefficients will have a non-standard unit of $[\frac{rad}{Nmm}]$, presenting values of different orders of magnitude smaller with respect to the other parameters. This discrepancy leads to the need to carefully choose the step size of the derivatives comprising the Jacobian identification matrix.

4.6. Sensitivity Analysis

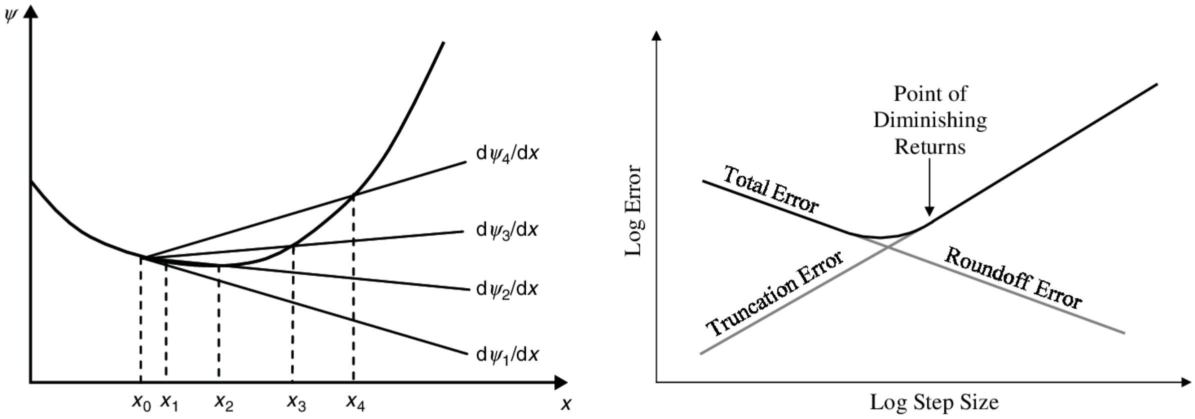


Figure 4.6: On the left, the influence of the step size in the example of the forward finite difference method. On the right, the effect of the step size on the numerical error.

The adjective "small enough" does not exactly convey the requirements the differentiation step size must respect, and when choosing it, several considerations must be taken into account to ensure its size does not lead to numerical errors:

- **Approximation Accuracy**: ε must be small enough so that the finite difference formula provides a good approximation of the derivative. If it is too large, the approximation error will be significant because the finite difference will not accurately represent the local behaviour of the function, as shown in the left frame of figure 4.6

- **Numerical Precision**: ε must not be so small that it causes significant numerical errors due to floating-point precision limits. Computers represent numbers with finite precision, and very small differences can lead to a loss of significant digits, introducing round-off errors. Considering the forward kinematics function is comprehensive of multiple matrix multiplications, the propagation of initially insignificant floating point errors may lead to important discrepancies in the results.

By manually tweaking the step size, it was immediately noticeable how the norm of the identification Jacobian matrix would change considerably each time. To solve this issue, a sensitivity analysis on the value of ε was needed.
Iterating for values of ε ranging from 10^{-7} to 10^{20}, and calculating the norm of the first row of the Jacobian corresponding to the default home position with all the joint angles null, the graphs shown below are produced.
Among the five options listed below that will be used for the next steps of the calibration, the sensitivity analysis was conducted on the first three:

- **OPTION 1** (28 parameters): Elasto-Kinematic Correction
- **OPTION 2** (21 parameters): Kinematic Correction
- **OPTION 3** (7 parameters): Elastic Correction
- **OPTION 4** (27 parameters): Elasto-Kinematic Correction without the k_0 parameter
- **OPTION 5** (6 parameters): Elastic Correction without the k_0 parameter

It is immediately clear how the plots of options 1 and 3 overlap almost perfectly, with values of the norm ranging around $7 \cdot 10^5$ while the purely kinematic model presents values three orders of magnitude smaller. Considering this enormous difference, the first observation is that a dedicated sensitivity analysis is needed for the purely kinematic and elastic cases.
This theory is also validated by the singular values of the Jacobian matrix, which are shown in Appendix A. As expected, the values are 28, mirroring the rank of the matrix, but what is concerning is the huge order of magnitude difference between the first and the last values. The first seven singular values are the consequence of the elastic coefficients, ranging between 10^3 and 10^6. Compared to the lower threshold of 0.95, it is a 3 to 6 difference in the order of magnitude. This effect is caused by the steep relationship between perturbations in the elastic coefficients and errors in the position of the end effector, which is clear when analysing the partial derivatives in the identification Jacobian matrix. This discrepancy then leads to the norm of the Jacobian being 3 orders of magnitude higher when dealing with the elastic coefficients. If the sensitivity analysis were to be conducted only for option 1 (the elastic-kinematic case, with 28 parameters), it would be impossible to understand the effect of the step size on the finite differences of the kinematic parameters, as the magnitude of the derivatives of the elastic coefficients would conceal the derivatives of the kinematic parameters. One more insightful consideration is that such a significant difference between the first and last singular values is a sign of an ill-conditioned matrix, which may lead to numerical instability when running the mathematical operations necessary for the identification phase. In case this phenomenon becomes critical and affects the results, the two calibration processes would be split into kinematic and elastic, so options 2 and 3.
To determine an appropriate step size, the sensitivity analysis involves calculating the norm of the Identification Jacobian Matrix constructed using various step size values. It has been empirically observed that a useful range lies between 10^{-15} and 10^{-3}. Step sizes above 10^{-3} exhibit noticeable truncation errors, whereas those below 10^{-15} consistently demonstrate

evident roundoff errors, as anticipated. This is visible in the plots shown on the next pages. Figure 4.7 compares the sensitivity analysis of three different options: the elasto-kinematic calibration, the purely kinematic calibration, and the strictly elastic calibration.

When numerical errors do not affect the norm's value, it is expected to remain constant. The plots indicate that this constancy occurs only within a specific range of step-size values. For the elasto-kinematic option (Option 1), this range is approximately 10^{-13} to 10^{-4}, while for the kinematic option (Option 2), it falls between 10^{-10} and 10^{-4}. As anticipated, due to the differing magnitudes of the Jacobian norm in elastic and kinematic cases, the elasto-kinematic case closely aligns with the elastic case. Consequently, the final plot exclusively compares the normalized kinematic and elastic cases. From this plot, it can be concluded that a safe step size is approximately between 10^{-10} and 10^{-4}.

These results were obtained while analyzing the arm in its home position, where all joint angles are set to zero. Testing different poses yields different results, indicating that the sensitivity analysis also depends on the arm's configuration. Figure 4.8 compares the results obtained with each joint set to 45 degrees on the left side and 90 degrees on the right side. Given the near-perfect overlap of the elasto-kinematic and elastic options, only the elastic and kinematic options are compared individually and then together in a normalized plot.

From the last normalized plots, it can be safely concluded that any step size between 10^{-10} and 10^{-7} would ensure numerical stability; therefore, a step size of 10^{-9} was selected.

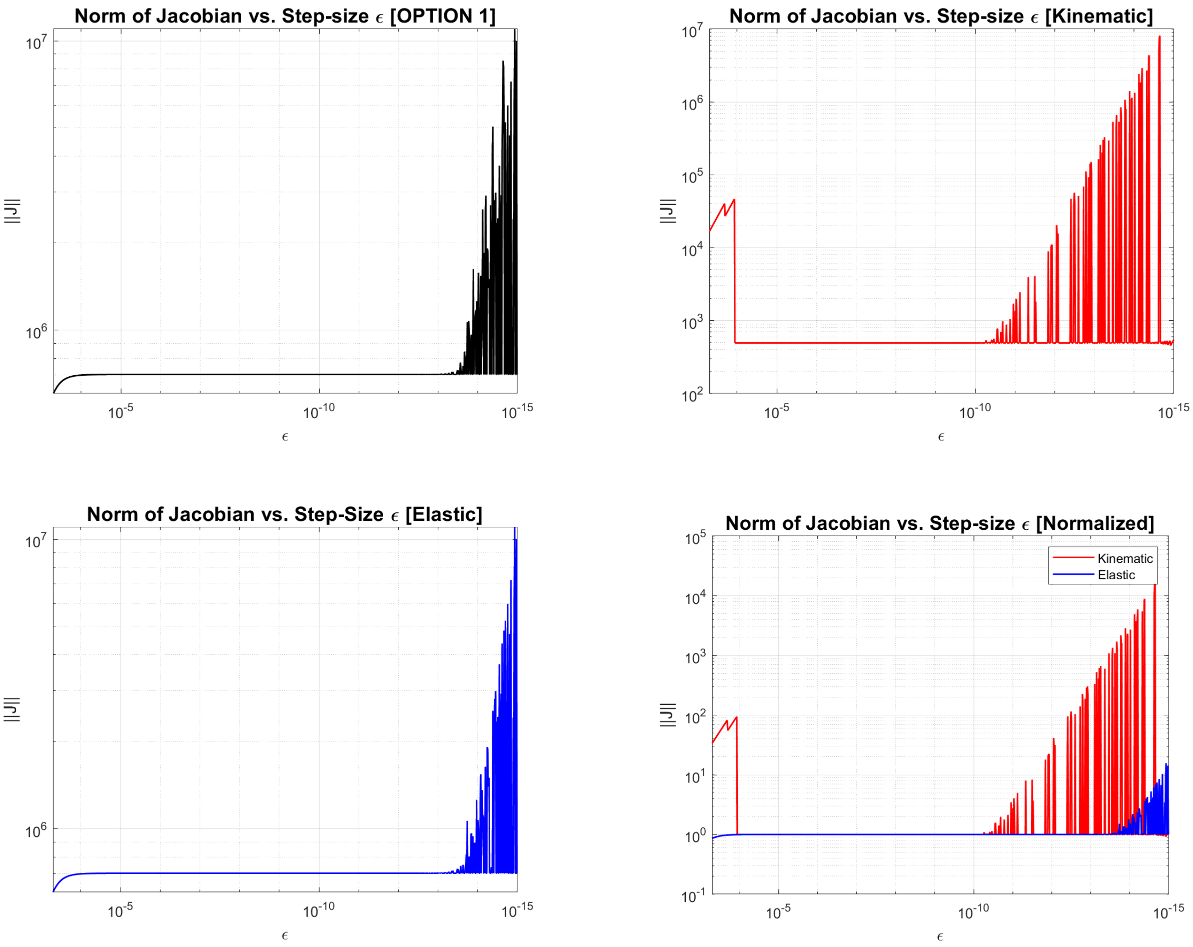


Figure 4.7: Comparison of the norm of the Identification Jacobian Matrix as a function of the central finite differences step size when each joint is at 0°, for the first 3 calibration options: option 1 (elasto-kinematic), option 2 (kinematic), option 3 (elastic).

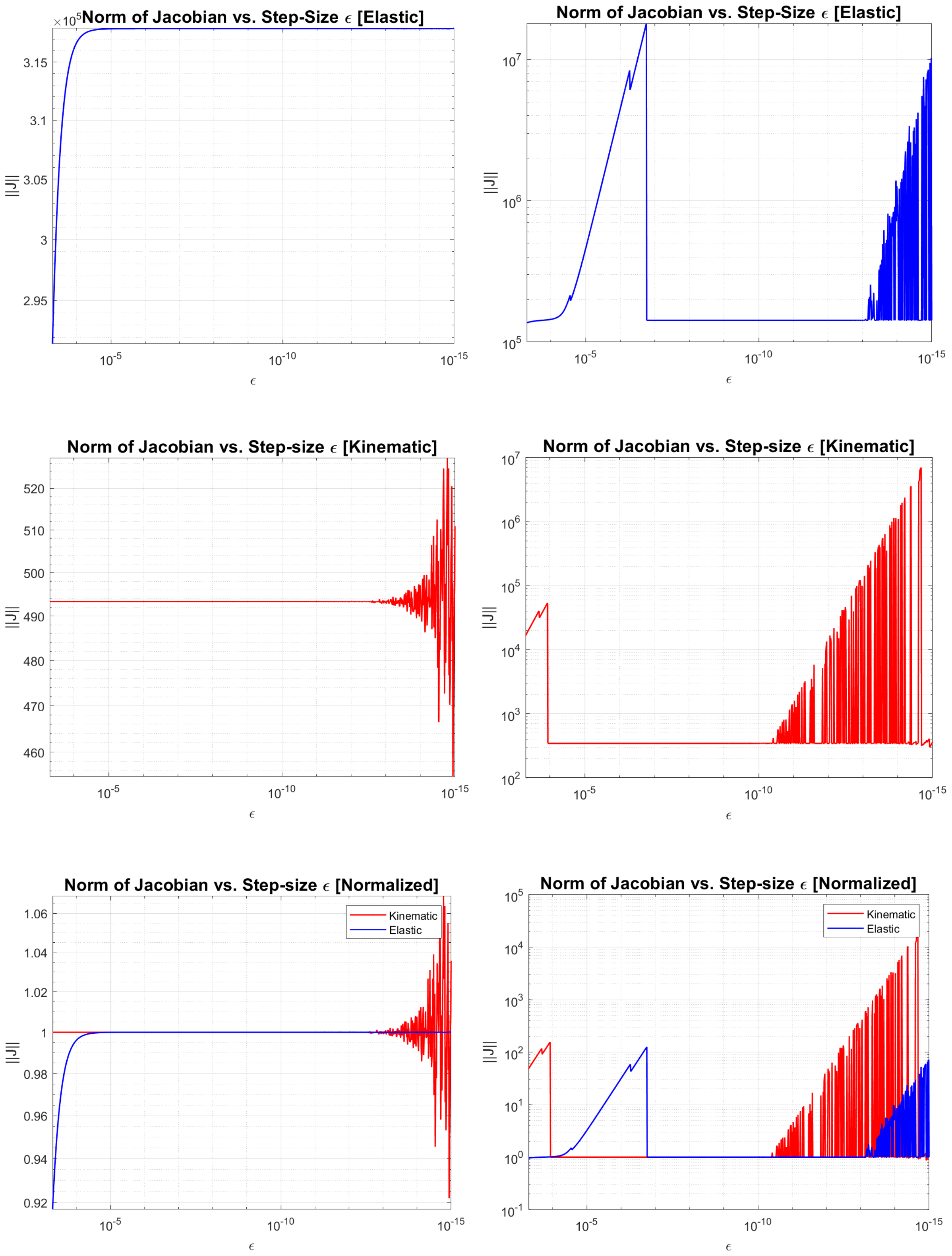


Figure 4.8: Comparison of the norm of the Identification Jacobian Matrix as a function of the central finite differences step size for the first 3 calibration options: option 1 (elasto-kinematic), option 2 (kinematic), option 3 (elastic). On the left, the results with each joint set at 45°. On the right, the results with each joint set at 90°.

5

Measurement

Once the model is ready, the next step is to measure the end effector's real position in different poses and compare it to the one expected by the kinematic model, obtaining a clear overview of the arm's positioning error. However, the challenge of retrieving precisely the needed data must not be underestimated. Several key factors must be addressed in planning this process, and each subsection will discuss them in detail: selecting the appropriate measurement system, accounting for potential self-intersection of the arm's links and collisions with the environment, correcting any misalignment between the arm's base and the measurement system's reference frame, and finally, establishing a measurement procedure that ensures accurate data collection while maintaining the arm's continuous operation.

5.1. The Advanced Realtime Tracking System

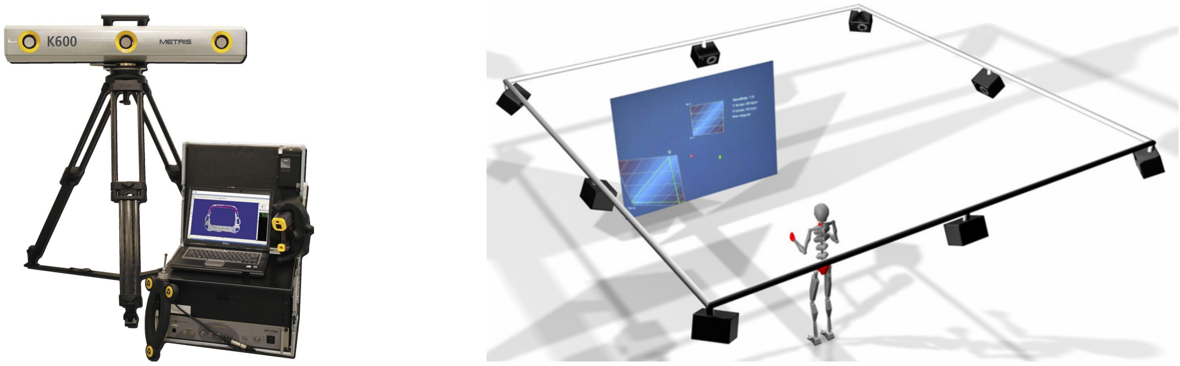


Figure 5.1: Different precise optical coordinate measurement systems. On the left, the METRIS K600, using active LED-based trackers. On the right, a render of the ART-Track system, which employs a set of cameras around the measurement area.

The Institute of Robotics and Mechatronics of DLR in Oberpfaffenhofen presents a variety of instruments able to capture the position of complex objects in real-time. The choice lay between the following different systems:

- **METRIS K-series**
  The METRIS K-series (METRIS, 2005) is a high-precision optical coordinate measurement system designed to capture objects' 3D position and orientation in real-time. It utilizes active markers, which are LED-based and emit infrared light at a customizable

frequency, allowing the system to detect them with high accuracy and speed. For accurate tracking, at least three markers are required to determine the position and orientation of a single object. These 2-3 cm in diameter active markers are strategically placed on the tracked object, enabling the system to provide precise feedback on its position and movement, weighing only a few grams. It excels in environments where accuracy and robustness are critical, as the active markers offer improved signal clarity and reduce the risk of interference from ambient lighting. This system can detect a wide range of geometries and is well-suited for dynamic applications where the object is in constant motion. The tracker consists of a portable machine equipped with wheels and three cameras, as shown on the left of Figure 5.1.
The METRIS K-series typically integrates with proprietary software developed by METRIS for data processing and analysis.

- **ART - Track**
  The ART (Advanced Realtime Tracking) Track system is a high-precision optical tracking solution designed to capture the 3D position and orientation of objects in real-time. It utilizes passive markers, which are small, lightweight, and coated with a retro-reflective material that reflects infrared light emitted by the system's cameras shown on the right of Figure 5.1. These markers are typically spherical, measuring around 1-2 cm in diameter, and weighing just a few grams, making them easy to attach to various objects without affecting their movement.
  To accurately track an object, the ART Track system requires at least four markers to determine its position and orientation. These passive markers are strategically placed on the tracked object, allowing the system to deliver precise feedback on its location and movement. The ART Track system is particularly effective in environments where high accuracy and real-time tracking are essential, as the passive markers are not affected by power constraints and offer consistent performance. On the other hand, the presence of reflective materials in the tracking scene may heavily affect its accuracy.

- **Vicon**
  Similar to the ART system, the Vicon tracker uses reflective markers that reflect infrared light from a network of high-speed cameras. The primary differences between the two systems lie in the software used to process the raw camera data into position and orientation information, as well as Vicon's capability to attach multiple reflective markers to dynamic shapes. This feature is especially advantageous in the film industry, where Vicon tracks actors' movements and translates them into computer-generated graphics.

Given the low weight of the arm, the need to measure static poses, and the requirement for high precision, the choice of measurement system was narrowed down between Vicon and ART. Active LED markers, such as those used by the METRIS K-series, were deemed too cumbersome due to their cables and setup requirements, making them less suitable for this application.
The ART system was selected over Vicon for several reasons. The ART system is already available in the Planetary Exploration Lab (PEL lab) at the Institute of Robotics and Mechatronics of DLR, an underground facility where I could conduct my experiments without interfering with other operations. This location offers the advantage of minimal disruption compared to using the Vicon system, which is located in a different lab where other rovers are tested.
Additionally, the ART system in the PEL lab has already been calibrated, simplifying its setup and operation as the position and orientation of the lab frame are known. Dark curtains control the lab environment, reducing reflections from metal beams, enhancing the accuracy of the measurements.

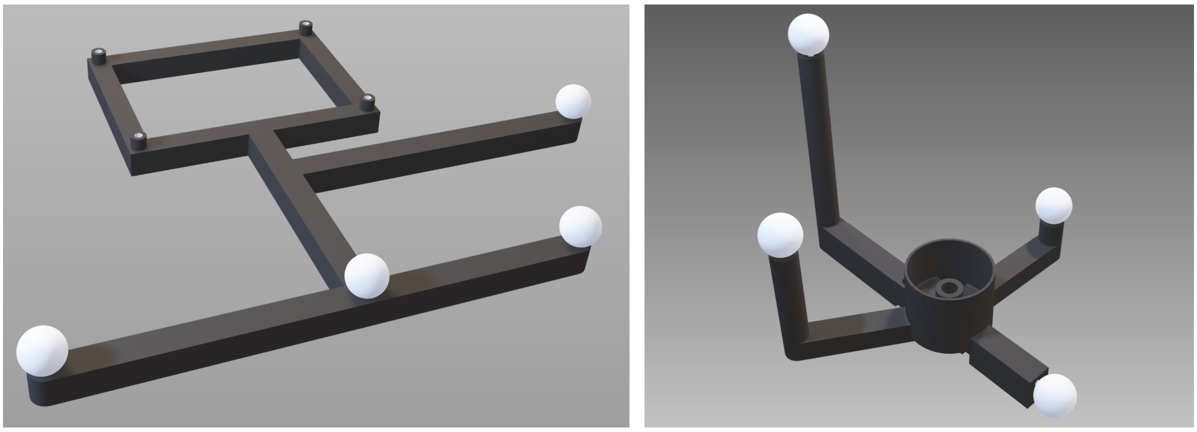


Figure 5.2: Different targets used for the ART-Track system, both using four spherical reflective markers. On the left, the one used to measure the position and orientation of the base frame, with all the markers on one plane, and with the right holes distances to precisely fit under the PCB controlling the arm. On the right, the target used to measure the position of the TCP of the robotic arm, with a socket designed to host heavy discs, and test the arm with different loads.

A dedicated tracking system is necessary to accurately determine the position of the arm's tip in relation to the base rather than just the lab's reference frame. Two different trackers were designed specially for the arm's measurements: a tracker for the end effector and one for the base of the arm. Both shown in Figure 5.2, present the minimum number of markers, which is four. The base tracker provides not only the position of the base within the lab's reference frame but also its orientation. This information is critical because the arm's kinematic model is built around the base, which is the origin. Without this tracker, the measurements of the tip would only be relative to the lab's reference frame rather than the base of the arm, leading to inaccuracies when comparing the real-world positions to the model's predictions. Also, if the base is not perfectly aligned with the pavement, the gravity gradient may not be perpendicular to its frame; therefore, a real estimation of the base orientation is necessary.

The markers on the trackers are positioned strategically to allow the system continuous detection. They are placed further than 1cm from each other and in random positions with respect to the reference frame. While in the base tracker, all the markers are on the same plane, as it is not expected to move during the measurement phase and therefore do not need to be always captured, the tracker for the end effector is designed differently. An increased distance between the markers means improved accuracy, however, an extremely elongated tracker would be counterproductive as it would exponentially increase the chances of self-intersection during the measurements. The result is a compromise between the maximum distance between the markers and compactness to avoid collisions. The position of the markers relative to the main reference frame of each target is presented in table 5.1

The tracker for the TCP was designed specifically to be able to eventually hold at a designated position, corresponding to a simple z translation with respect to the reference frame of the end effector, an external load. A diameter for the enclosure of 15 mm guarantees the possibility of using simple aluminium discs of 11 grams stacked on each other and kept in place by an M6 nut threaded onto an Allen key socket head screw. This way, it is possible testing the same poses with different loads at the end effector, simulating what would happen when picking up objects such as rocks and instruments.

Base Target				TCP Target		
n°	x[mm]	y[mm]	z[mm]	x[mm]	y[mm]	z[mm]
1	245	0	17	52	0	-4.5
2	245	80	17	0	60	17
3	245	-100	17	-30	-50	47
4	175	125	17	-75	0	87

Table 5.1: Table displaying the coordinates of each marker used to track the base's position and orientation (on the left) and the TCP (on the right), relative to their respective coordinate frame origins.

5.2. Collision Avoidance

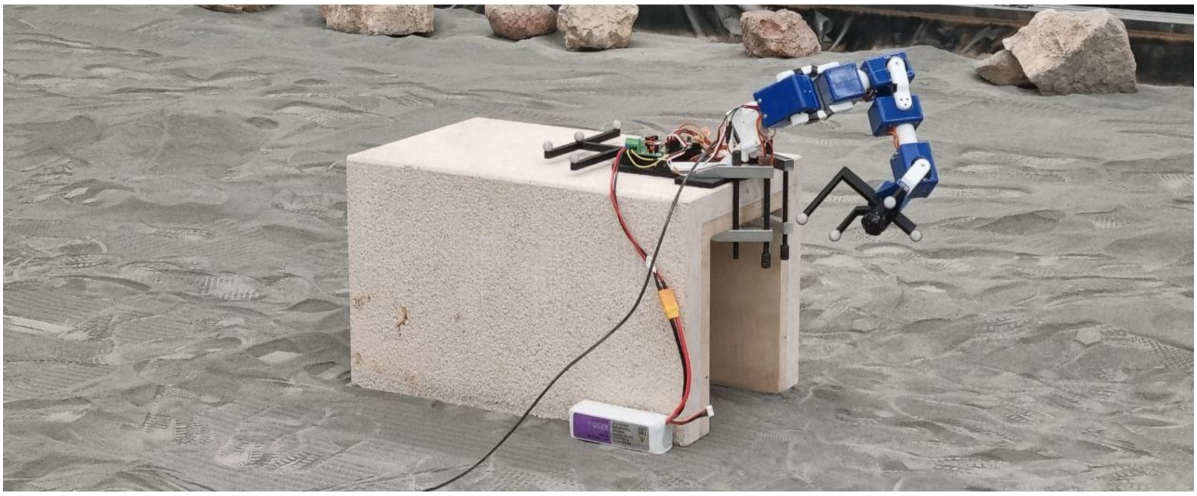


Figure 5.3: Picture of the setup used to measure the end-effector position in the PEL lab. The base of the arm is fixed to a heavy marble stone using a pair of clamps, which also keep the PCB and the base's target in place. A LiPo battery powers the setup.

The complete setup is shown in Figure 5.3, and will be described in section 5.4
It is clear that the presence of the marble stone holding the base of the arm firmly to avoid oscillations during movements will impact its reachability. Not only will certain poses not be possible as the marker would intersect the base, but during movements between one pose and the next one, it could still get dangerously close, and due to its flexibility, it could still impact the stone and get damaged. To solve this issue, the model was imported into MATLAB. The robotics toolbox allows the creation of a simple collision mesh by interpolating the external surfaces of each link, allowing to check for both self-intersections and collisions with the surrounding environment in each pose. The model, and the collision meshes are shown in figure 5.4.

To ensure safe operations, among the entire group of possible poses, only the ones respecting

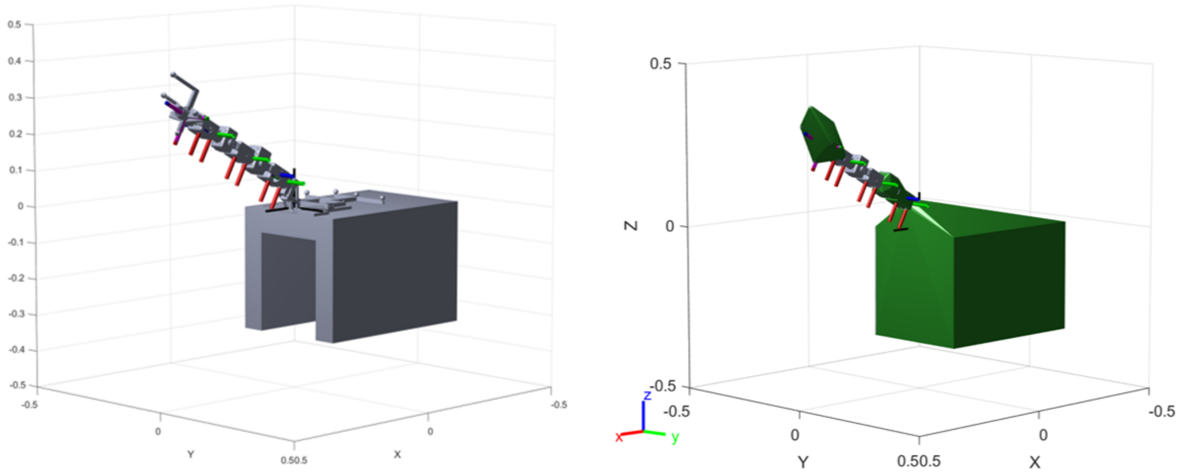


Figure 5.4: Picture displaying the whole measurement setup imported into Matlab Robotics Toolbox. The special ”collision” feature enables the creation of wrapping geometrical shapes around each joint, highlighted in green on the right. This allows detecting and avoiding possible collisions during the measurement process.

the following requirements were selected: The minimum distance between the second link and the base must be at least 3 cm; the end-effector, comprehensive of the target, must always be further than 16 cm from the rest of the arm and the marble base.
Respecting these requirements and considering three configurations for each joint corresponding to -100, 0, and 100 degrees, from a total of $3^6 = 729$ poses, the number of required measurements went down to 393. Only the three extreme positions were used mainly because four waypoints would have meant an unmeasurable total of 4096 positions, and then for two simple reasons. First of all, to cover the broadest range of possible end-effector positions, then to ensure the first calibration level would be adequate.
Once these poses have been identified, they are stored in a text file named ”poses.txt”.

5.3. Preliminary Tests

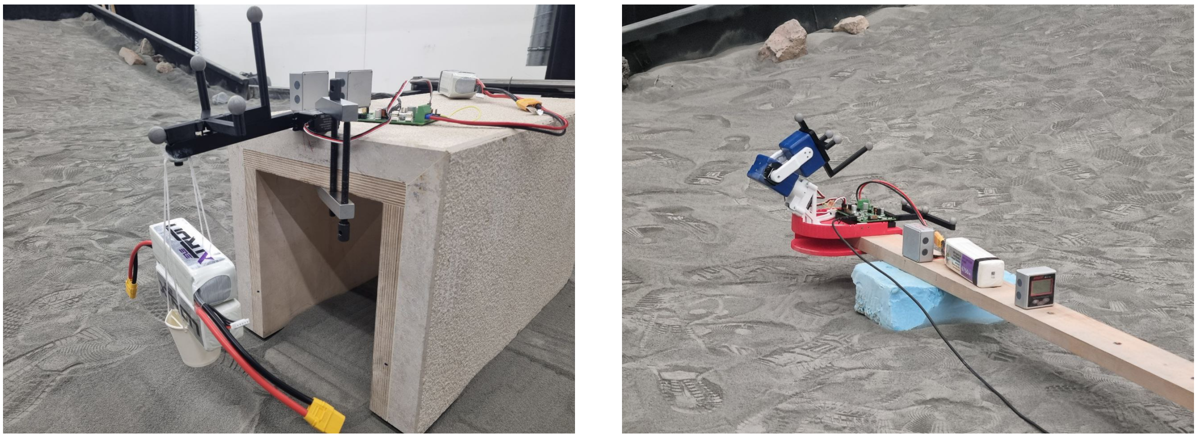


Figure 5.5: Pictures displaying the preliminary tests conducted in the PEL lab. On the left, the simple setup that allowed the identification of the order of magnitude of one servo torsional elastic coefficient. On the right, only the first two joints are tested to validate the overall methodology of the parameters identification process.

Before measurements were conducted on the entire arm, the real position of the end-effector target with respect to the base target was measured. Then, the servos were tested in two separate scenarios: initially, a single servo was measured under varying loads to estimate the order of magnitude of its elasticity. Finally the complete calibration methodology was trialled on the arm's first two degrees of freedom.

- **Alignment Base target - TCP target**
  Although both the targets have been particularly designed to minimise the localisation error of the ART-Track by using extended beams, the small size of the reflective markers may induce errors in the computation of the real origin of the reference frame. Also, the extended beams may experience increased deflection due to UV exposure and heat, bending as illustrated on the left in Figure 5.6, and affecting the detected position of the markers. To tackle this issue, a simple device was designed to keep the two targets at a known distance, and measure their real relative position. The setup is shown on the right of Figure 5.6.
  The result of this preliminary test is a deviation of the TCP target with respect to the Base Target of [-0.303 -0.692 0.434] mm.

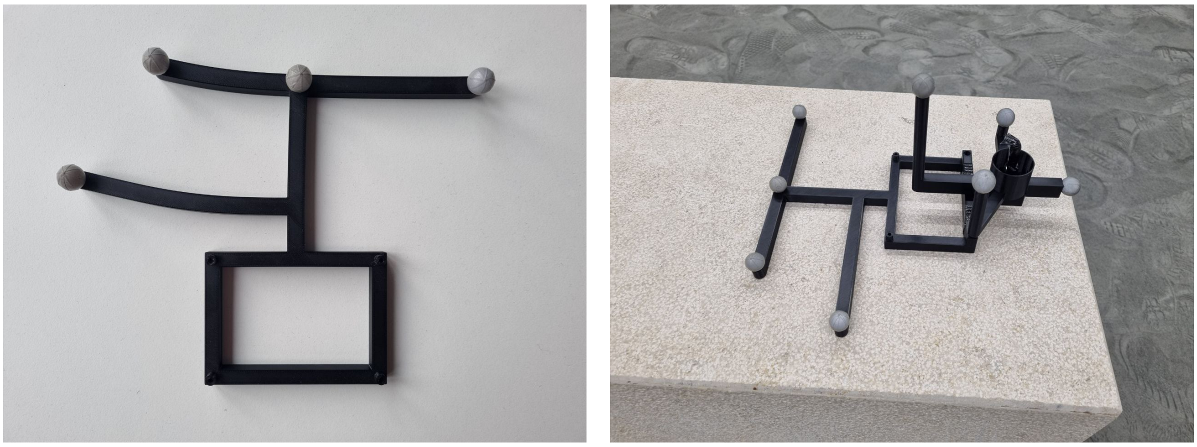


Figure 5.6: On the left, the effect of extended sunlight exposure on the Base Marker, bending the Polylactic Acid (PLA) beams and rendering the target unusable. On the right, the setup of the test conducted to measure the real position of the end-effector reference frame with respect to the one of the Base target.

- **One Joint**
  Modern literature lacks sufficient information on the elastic deformations of RC servos, largely because these motors are typically not used in applications requiring high precision. However, as explored in Chapter 2, there are some cases where elastic calibration of more complex robotic arms has been studied. The elastic coefficients identified in these studies range from $4.5 \cdot 10^{-5}$ rad/Nm (Khalil & Besnard, 2002) to $1 \cdot 10^{-2}$ (Tenhumberg & Bauml, 2021). To explore this aspect in the context of RC servos, a single servo was first tested, specifically the AGF A80BHMF. The servo was securely clamped in place, with a structure attached to its head to apply the load. While the setup might appear rudimentary, the simplest and lightest way to suspend a specific weight was by using a paper cup. LiPo batteries were employed as weights, carefully hung from the servo to measure its elastic response under three different loads corresponding to one, two and three batteries. To measure the different inclinations, the same target designed for the arm end effector was secured in place at the tip of the structure screwed into the servo. The result is shown in Figure 5.7 and depicts a behaviour very close to being linear, with slope values ranging from $2.36 \cdot 10^{-2}$ to $3.42 \cdot 10^{-2}$ rad/Nm when forcing the servo to torques up to 2.6 Nm. These results are slightly outside the range of the coefficients found in the referenced literature, which is understandable given the lower cost

and construction quality of RC servos compared to the more advanced motors used in those studies.

- **2 Degrees of Freedom** The next test was designed to validate the methodology, verify the functionality of the camera system, and confirm all procedures before advancing to the more complex 6 degrees of freedom calibration. To achieve this, a slightly modified target was created to fit the second joint while maintaining the same relative distances between the spherical markers to avoid the need for re-calibration or the integration of a new target in the ART system. This test demonstrated that the ART system could successfully capture both the TCP and base targets simultaneously, with noise levels below half a millimetre. A total of 100 poses were measured; Figure Figure 5.7 shows the 3D position of the TCP for the first 14 measured poses. Each step corresponds to a different pose, and the spikes visible before the arm reaches the 11th pose are caused by the simultaneous movement of the first joint from -100° to -60° and the second joint from +100° to -100°. This behaviour arises from the selection of poses, which totals 100 as a result of 10^2 configurations. Each joint cycles through 10 distinct positions, so every ten poses, the second joint always moves from +100° to -100°, causing the observed spike.

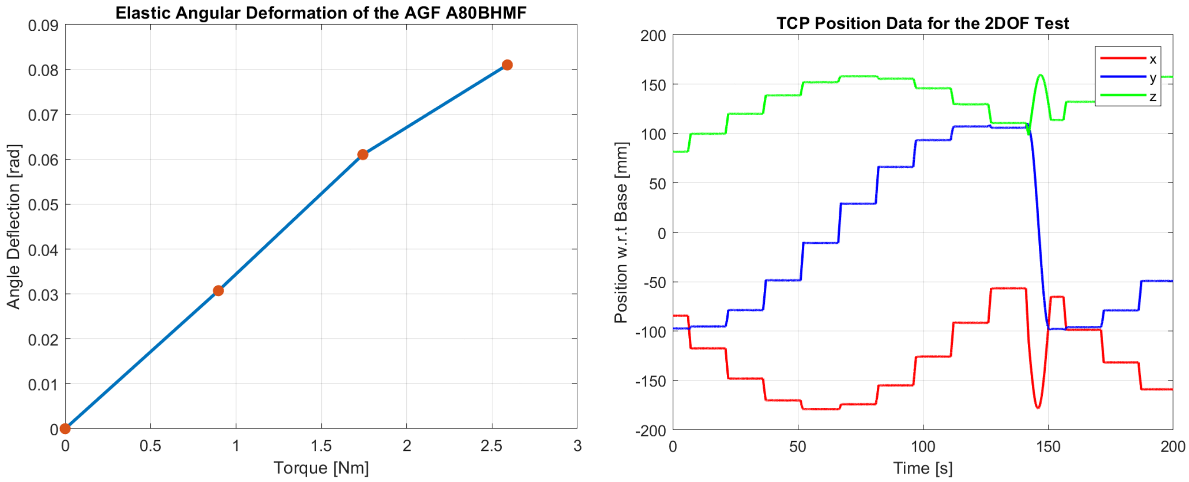


Figure 5.7: On the left, the result of the first test conducted on one servo, showing the almost linear dependency of the rotational elastic coefficient, with values ranging from $2.36 \cdot 10^{-2}$ to $3.42 \cdot 10^{-2}$ rad/Nm. On the right, an example of the data retrieved thanks to the ART-Track system, showing the x, y, and z position of the end-effector target in the first 14 poses among the 100 recorded.

5.4. Measurement Setup

The setup has already been shown in Figure 5.3. The arm is screwed into a black 3D-printed base which mirrors the screw configuration used in the rover. This base is clamped with two metal clamps into a marble stone, keeping the entire arm in place and heavily limiting any possible disturbance throughout the measurement process. With respect to the rover configuration, only one cover is missing to facilitate the movement of cables. Each servo is connected to the custom PCB taken directly from one of the rovers and connected to a LiPo battery and a PC via USB.

Although only a certain number of poses is necessary for the identification phase, it was still decided to measure the arm in all possible configurations to have some spare data useful for the validation phase.

From past experiences, it was decided that the maximum rotational speed to ensure safe operations of the arm and avoid the links breaking due to the high inertia, was 20 degrees per second. This means that to allow the arm to reach every single pose safely, the minimum time lapse that must occur between two opposite configurations, which means with a joint going from -100° to +100°, is 12 seconds. However, due to its elasticity, the arm is expected to oscillate briefly when reaching the position before getting stable. To ensure correct data gathering and to leave a safety margin, the timer responsible for changing the target pose is set to tick every 15 seconds.

When extrapolating the data that will be analysed in the following phases, the position measured in that time frame will be determined by an average of the measurements collected at the last second of each loop.

The methodology is shown in Figure 5.9. As previously mentioned, the file outputted by the collision avoidance algorithm is named ”poses.txt”, and contains all the target angles of each joint. While each line is read every 15 seconds, each joint moves slowly to the target position thanks to the addition of values of 1 or -1 every 6.5 milliseconds. These values are not straight angle values, but microseconds of the PWM signal already introduced in chapter 3. For example, moving the first joint from the minimum value of -100° corresponding to a PWM signal of 620 μs, to +100° which is 2460 μs, it would take $(2460-620)\cdot 0.0065 = 11.96$ seconds, close to the 12 seconds aforementioned.

When this methodology was first tested, the results were promising. The arm would easily reach every position indicated in the poses.txt file, and in around 100 minutes, it would complete the measurements. However, the idea behind this calibration is to test the arm with different loads at the end effector and compare the results of the identification phase, proving the linearity of the joint elasticity or at least having two different sets of calibration parameters for the target approaching and retrieving phases. To do so, it was necessary to add a load to the socket in the target at the end effector; however, after around 30 minutes of measurements, one joint failed. Upon further inspection, the reason for its failure is shown in Figure 5.8. Due to the high torque needed, the fourth joint overheated.

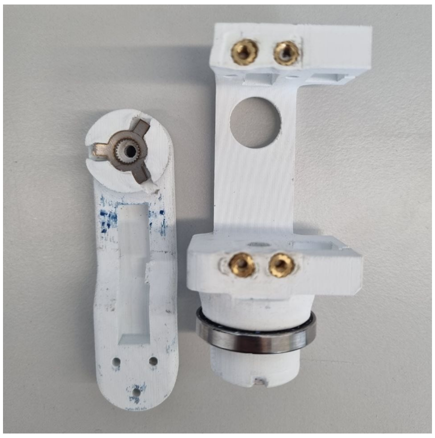


Figure 5.8: Picture of the components affected by the meltdown of one of the servos during the first test runs. On the left, it is clear how the melting ASA allowed the inserts to slip out of the reserved holes and the damage at the base of the servo's wiring. On the right, the damage to the other component is more clear, with the huts designed for the servo metallic heads fatally worn out.

The effects of the high temperature reached are clear in the inserts used to keep the servo in place, as well as in the steel insert screwed into the servo head. The ASA plastic is completely melted, indicating the servo reached temperatures higher than 150° C (Flashforger, 2024). When trying to mount the motor again, it would simply not work, indicating that some internal component was damaged, probably due to a short circuit within the windings.
The servo and plastic components were replaced to fix this issue, and the measurements were divided into six shorter batches.

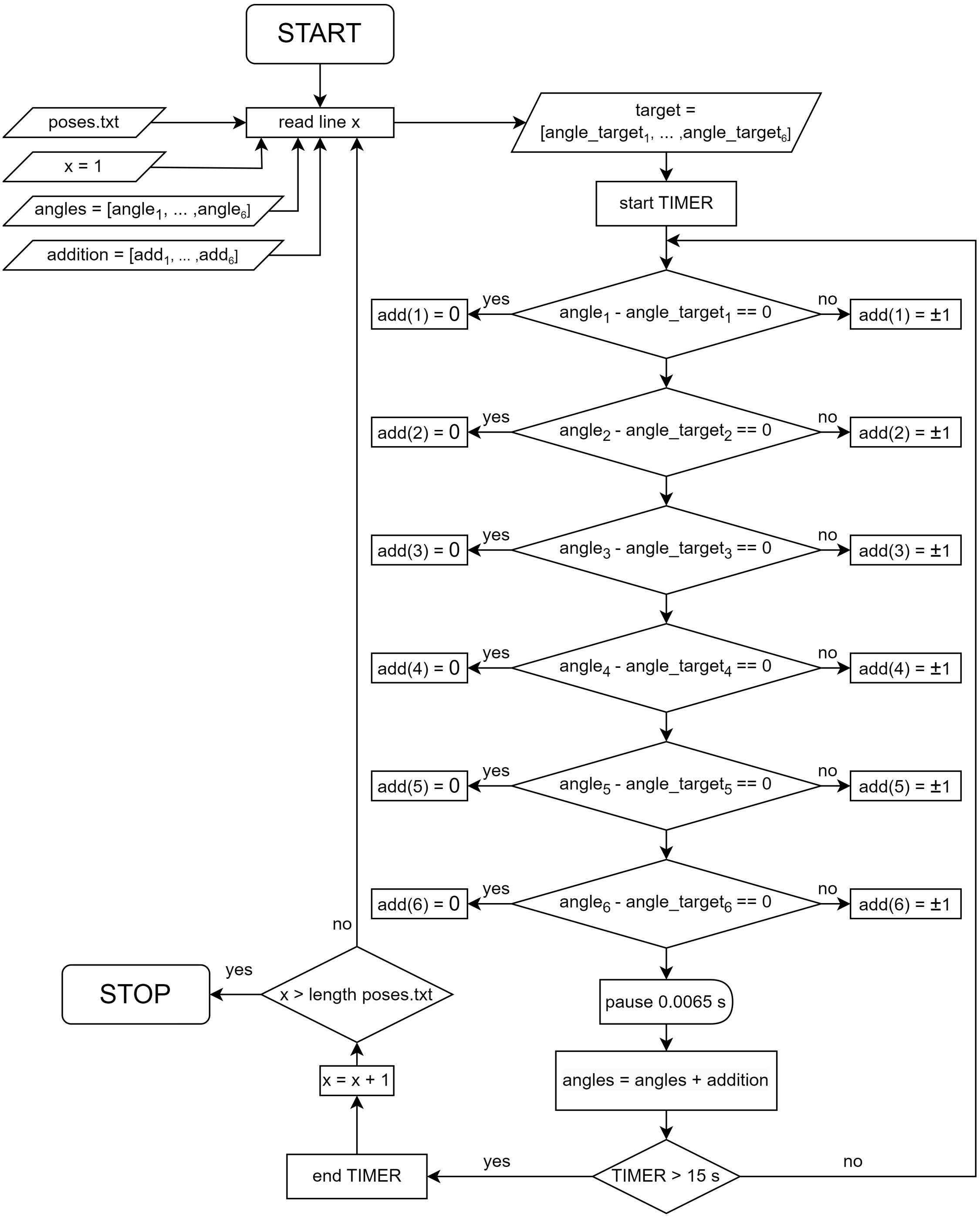

START
poses.txt
x = 1
angles = [angle1, ... ,angle6]
addition = [add1, ... ,add6]
read line x
target = [angle_target1, ... ,angle_target6]
start TIMER
angle1 - angle_target1 == 0
yes
add(1) = 0
no
add(1) = ±1
angle2 - angle_target2 == 0
yes
add(2) = 0
no
add(2) = ±1
angle3 - angle_target3 == 0
yes
add(3) = 0
no
add(3) = ±1
angle4 - angle_target4 == 0
yes
add(4) = 0
no
add(4) = ±1
angle5 - angle_target5 == 0
yes
add(5) = 0
no
add(5) = ±1
angle6 - angle_target6 == 0
yes
add(6) = 0
no
add(6) = ±1
pause 0.0065 s
angles = angles + addition
TIMER > 15 s
yes
no
end TIMER
x = x + 1
x > length poses.txt
yes
no
STOP

Figure 5.9: Flowchart of the process followed to move the arm from one pose to the next one, with an interval of 15 seconds.

6

Identification and Validation

This chapter will analyse the last phases of the calibration process: identifying the parameters described in the model section, validating them, and applying them to the forward kinematics. If the model is correct, these parameters should improve the quality of the arm's kinematics and decrease the error in the end effector position calculated by comparing the data obtained in the previous measurement phase with the forward kinematics. The five parameter sets already introduced will be considered, and their performance will be evaluated and compared.

6.1. Measured Data Analysis

As anticipated in the previous section, the data retrieved will have a very similar shape to the one presented in Figure 5.7. The total number of variables gathered is 7, consisting of 3 variables for the end effector's position and 4 for its orientation. The orientation is represented using quaternions, which means the four variables define the orientation in a four-dimensional space.

A quaternion is given by:

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{6.1}$$

where w, x, y, z are the quaternion components, and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the imaginary units. The quaternion provides a compact and computationally efficient way to represent rotations and avoid issues like gimbal lock, which is common with other representations such as Euler angles, and it is the default output format of the ART system.

Once these variables are collected for each pose, following the same schematic employed in the previous section for the 2 DOF test, they must be adjusted to be compared to the forward kinematics with the following procedure:

1. The position of the base target is subtracted from each component of the end-effector's position. Since the base target shares the same origin as the reference frame in the forward kinematics, no additional translations are required.

2. The position vector of the end effector is then rotated using the orientation data from the base target to align it with the forward kinematics.

6.2. The Observability Index

Once the measured data is collected, the next step is identifying which of the 393 poses should be used to determine the parameters. As explained in chapter 2, a safe number of poses for such a task would be 50. The challenge is identifying which group of 50 poses will most likely determine the parameters that best align the forward kinematics to the measured 393 poses. To do so, a valuable instrument is the observability index. Although different options for the observability index exist, the one identified as the best performing by Joubair and Bonev (2013) is here reported:

$$O = \frac{\sqrt[m]{\sigma_1 \sigma_2 \cdots \sigma_m}}{\sqrt{(n)}} \tag{6.2}$$

This index is calculated from the SVD of the Jacobian Identification Matrix of Eq 2.3, ensuring that the selected poses offer the most comprehensive and distinct information possible, making the parameter identification more robust. Typically, the approach involves selecting the group of 50 poses that maximizes this index. One effective way to tackle this task is through an optimal design search algorithm, with DETMAX (Mitchell, 1974), or D-optimal design via exchange method, being among the most widely used. The workflow of this algorithm is shown in Figure 6.1. It consists of adding the one configuration that maximises the observability index to an initial random set of 50 poses, leading to 51 poses. Then, delete the one that maximises the remaining 50 poses and repeat this procedure until the algorithm adds and deletes the same pose.

Considering the whole process is based on the Identification Jacobian Matrix, it is clear that the set of poses depends on which parameters are considered to build the matrix. For this simple reason, each option listed in section 4.6 will have its own set of poses which maximise the observability index.

In this context, the DETMAX algorithm proves especially useful in handling large datasets, such as the 393 poses measured in this experiment, by reducing the selection to a manageable and optimal subset of 50 poses, denoted as $\mathbf{\Psi}$. The iterative approach of adding and removing poses based on the observability index ensures that the final set of poses provides the most distinct and valuable information for parameter identification. This not only improves the accuracy of the forward kinematics model but also reduces the computational complexity of the parameter estimation process.

Another important consideration is the effect of different parameters on the Identification Jacobian Matrix. As previously mentioned, each option listed in section 4.6, which includes varying elasto-kinematic and kinematic models, results in a different set of parameters. Consequently, the structure of the Jacobian matrix changes depending on the parameters being considered. This implies that the observability index and the optimal pose set may differ for each option, requiring a tailored application of the DETMAX algorithm for each case.

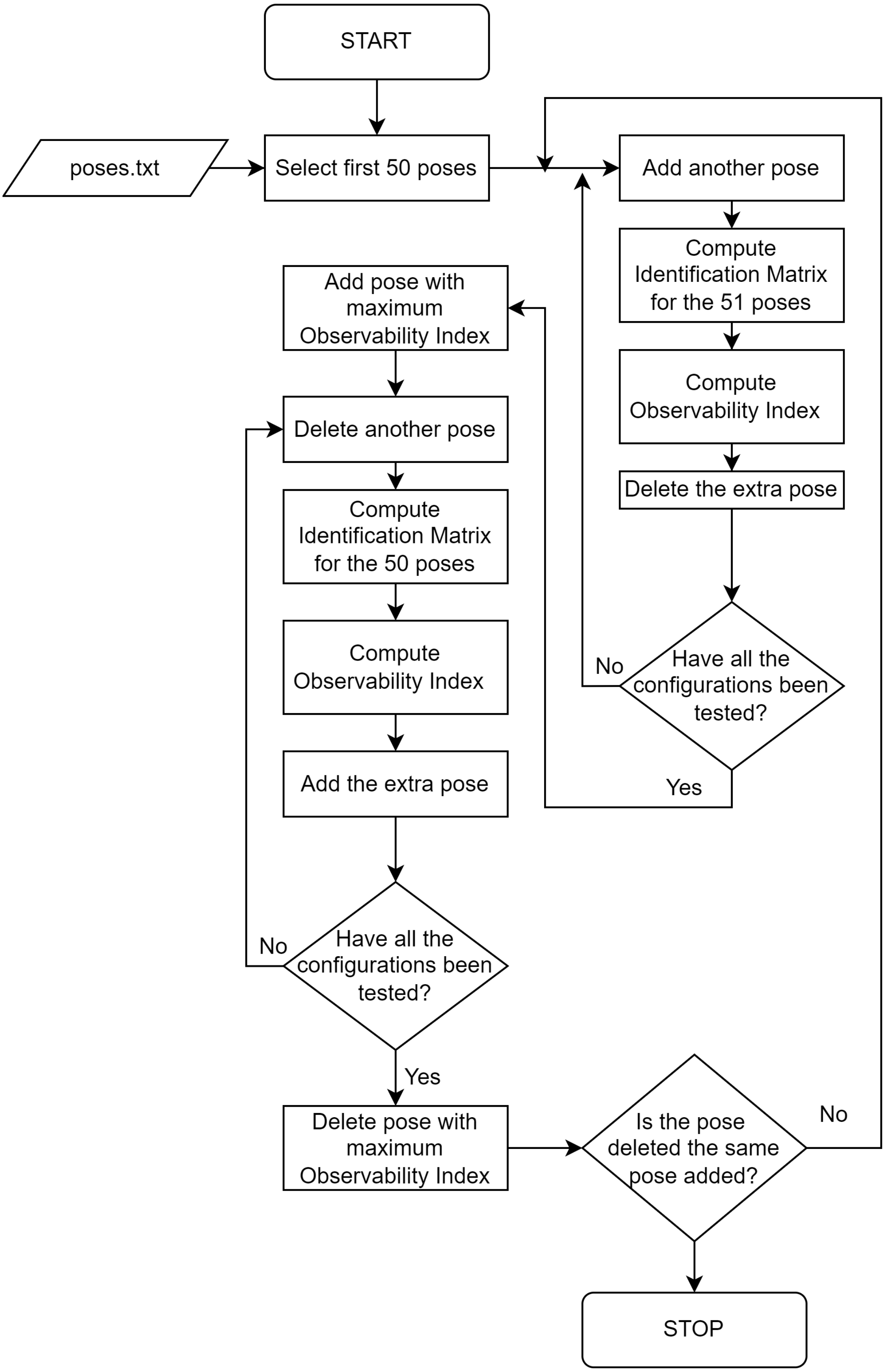

START
poses.txt
Select first 50 poses
Add another pose
Compute Identification Matrix for the 51 poses
Compute Observability Index
Delete the extra pose
Have all the configurations been tested?
No
Yes
Add pose with maximum Observability Index
Delete another pose
Compute Identification Matrix for the 50 poses
Compute Observability Index
Add the extra pose
Have all the configurations been tested?
No
Yes
Delete pose with maximum Observability Index
Is the pose deleted the same pose added?
No
STOP

Figure 6.1: Flowchart of the DETMAX algorithm necessary to isolate the best set of poses among the ones measured, for the identification phase.

6.3. Parameters Identification

As already mentioned, five different options were considered for the identification process:

- **OPTION 1** (28 parameters): Elasto-Kinematic Correction
- **OPTION 2** (21 parameters): Kinematic Correction
- **OPTION 3** (7 parameters): Elastic Correction
- **OPTION 4** (27 parameters): Elasto-Kinematic Correction without the k_0 parameter
- **OPTION 5** (6 parameters): Elastic Correction without the k_0 parameter

As mentioned in Chapter 2, the identification process is based on a least squares method, which adjusts the parameter values to fit the measured data. A schematic is provided in Figure 6.2. The process begins by selecting one of five possible options, along with the set of poses, $\mathbf{\Psi}$, which were determined in previous studies of the observability index. After importing the measured data, the algorithm enters a while loop.

In the first step of the loop, the parameter vector $\boldsymbol{\nu}$ is initialized as a zero vector, which is then used to compute the identification Jacobian matrix $\mathbf{J}$. Using this initial vector $\boldsymbol{\nu}$, the estimated positions of the end effector are calculated.

The difference between the measured positions of the end effector and the estimated positions from the previous step is used to compute the error in $\boldsymbol{\nu}$, as shown in the equation below:

$$\Delta\boldsymbol{\nu} = \mathbf{J}^{+} \cdot \Delta\mathbf{X} \tag{6.3}$$

with $\mathbf{J}^{+}$ the pseudo-inverse of the Jacobian Identification Matrix. The vector $\boldsymbol{\nu}$ is then updated based on this computed error, and the cycle repeats. This process continues until the ratio between the norms of $\Delta\boldsymbol{\nu}$ from two consecutive iterations drops below a certain threshold, which has been experimentally determined to be around $1 \cdot 10^{-9}$.

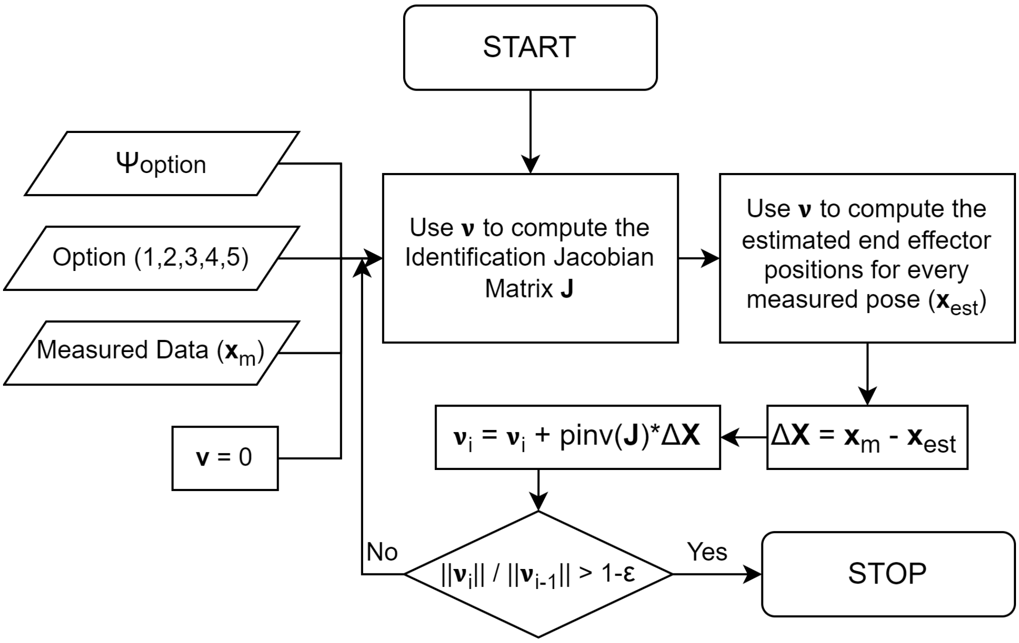

START
Ψoption
Option (1,2,3,4,5)
Measured Data (xm)
v = 0
Use v to compute the Identification Jacobian Matrix J
Use v to compute the estimated end effector positions for every measured pose (xest)
ΔX = xm - xest
vi = vi + pinv(J)*ΔX
||vi|| / ||vi-1|| > 1-ε
No
Yes
STOP

Figure 6.2: Flowchart of the least-square identification method. It begins with importing the measured data and the correct $\mathbf{\Psi}$, choosing the right option, and ends when the vector $\boldsymbol{\nu}$ of parameters converges.

Although the comprehensive results of the identification are presented in Appendix B, table 6.1 highlights the most significant parameters for each option.

Parameter	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Option 5**
k_0 [rad/Nm]	1.93e-2	0.00	2.23e-2	0.00	0.00
k_1 [rad/Nm]	7.56e-2	0.00	7.76e-2	7.88e-2	9.32e-2
k_2 [rad/Nm]	6.70e-2	0.00	8.24e-2	8.66e-2	9.55e-2
k_3 [rad/Nm]	7.68e-2	0.00	9.32e-2	8.94e-2	1.14e-1
k_4 [rad/Nm]	1.24e-1	0.00	1.473e-1	1.27e-1	1.41e-1
k_5 [rad/Nm]	1.04e-1	0.00	-4.84e-3	3.69e-1	5.42e-1
k_6 [rad/Nm]	-7.39e-3	0.00	6.27e-1	1.96e-1	6.23e-1
δz_b [mm]	-5.31	-25.9	0.00	-6.44	0.00

Table 6.1: List of the most relevant parameters, and the relative results of the identification process for each option.

These parameters were chosen for two main reasons. First, the elastic coefficients are the most relevant to the topic of this thesis, the elastic compensation of the arm joints. Second, the kinematic parameter with the most significant magnitude is the translation in the z direction of the base.

From a first analysis, it is clear that for options 1 and 3, the elastic coefficient of the fourth joint is negative.

Lastly, it is interesting to notice how the displacement of the base position along the vertical z-axis is heavily dampened by the presence of the elastic coefficients, shifting from more than 2 cm to 5 and 6 mm.

The presence of non-physically acceptable parameters is not due to errors in the methodology but is likely a result of the model's lack of detail. One solution is to increase the model's complexity by adding more independent parameters, though this might not fully resolve the issue.

A more straightforward approach would be to apply constraints within the least squares method to ensure the parameters remain physically realistic.

Noticing that the result of the identification phase is heavily dependent on the set of poses used to determine the parameters, to force the constraints it is sufficient to add one more step to the DETMAX algorithm. The solution is shown in Figure 6.3. Adding one more step after computing the observability index of each set of poses allows running an inter-loop identification. If the results of this process are non-physically acceptable (e.g. the elastic coefficients are negative), the set is discarded by assigning an observability index value of 0.

As shown in Table 6.2, a physically feasible solution exists for both options, with only positive elastic coefficients.

Parameter	$k_0[\frac{\text{rad}}{\text{Nm}}]$	$k_1[\frac{\text{rad}}{\text{Nm}}]$	$k_2[\frac{\text{rad}}{\text{Nm}}]$	$k_3[\frac{\text{rad}}{\text{Nm}}]$	$k_4[\frac{\text{rad}}{\text{Nm}}]$	$k_5[\frac{\text{rad}}{\text{Nm}}]$	$k_6[\frac{\text{rad}}{\text{Nm}}]$
Option 1_2	1.81e-2	7.71e-2	7.25e-2	8.22e-2	1.21e-1	2.17e-1	8.06e-3
Option 3_2	2.40e-2	7.75e-2	8.04e-2	9.05e-2	1.40e-1	9.51e-2	5.08e-1

Table 6.2: Table showing the results obtained with the modified identification process, which considers the system's physical constraints: the positivity of the elastic coefficients.

In order to prove the linearity of the elastic coefficients shown in Figure 5.7, a second round of measurements was conducted, attaching a load of 66.3 g at the designed location on the target of the end-effector. The results of the identification process are shown in the table below. Interestingly, all five options yield physically valid solutions and reveal a notable trend in the elastic coefficients absent in prior results. Specifically, the first three servos exhibit a lower order of magnitude in elasticity than the others. This could be attributed to the different servo models used, as the servos controlling the first three degrees of freedom are larger and sustain more weight, therefore likely less elastic.

The next step is to demonstrate that these coefficients hold even in the absence of an applied load, which would confirm that the elasticity of the joints behaves linearly.

Parameter	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Option 5**
k_0 [rad/Nm]	2.09e-2	0.00	2.29e-2	0.00	0.00
k_1 [rad/Nm]	7.11e-2	0.00	7.36e-2	7.34e-2	8.85e-2
k_2 [rad/Nm]	5.96e-2	0.00	7.45e-2	7.73e-2	8.70e-2
k_3 [rad/Nm]	6.56e-2	0.00	8.16e-2	7.88e-2	9.83e-2
k_4 [rad/Nm]	1.14e-1	0.00	1.35e-1	1.19e-1	1.32e-1
k_5 [rad/Nm]	1.29e-1	0.00	3.06e-2	3.83e-1	5.58e-1
k_6 [rad/Nm]	1.20e-1	0.00	4.39e-1	1.92e-1	5.09e-1
δz_b [mm]	-5.43	-29.4	0.00	-7.43	0.00

Table 6.3: List of the most relevant parameters, and the relative results of the identification process for each option, in the case of a load of 66.3 g applied at the end-effector.

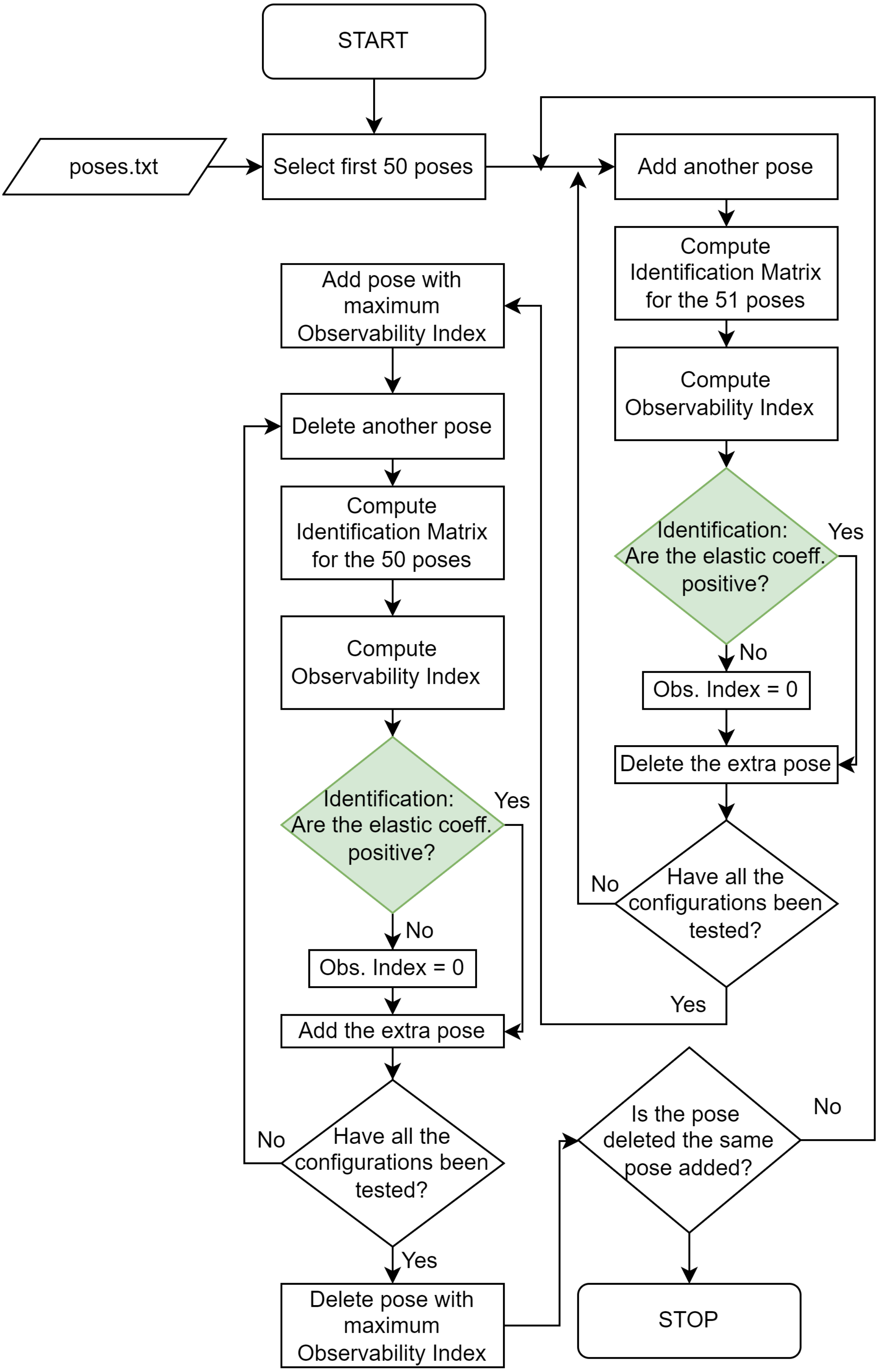

START
poses.txt
Select first 50 poses
Add another pose
Compute Identification Matrix for the 51 poses
Compute Observability Index
Identification: Are the elastic coeff. positive?
Yes
No
Obs. Index = 0
Delete the extra pose
Have all the configurations been tested?
No
Yes
Add pose with maximum Observability Index
Delete another pose
Compute Identification Matrix for the 50 poses
Compute Observability Index
Identification: Are the elastic coeff. positive?
Yes
No
Obs. Index = 0
Add the extra pose
Have all the configurations been tested?
No
Yes
Delete pose with maximum Observability Index
Is the pose deleted the same pose added?
No
STOP

Figure 6.3: Flowchart of the modified identification process, including the physical constraint of positive elastic coefficients.

6.4. Validation

As mentioned in the previous Chapters, the total number of poses measured is 393, while the ones used for parameter identification are only 50. This means that the remaining 343 poses can be used as an insightful tool to validate the whole methodology and evaluate how the calibration process increases the precision of the end-effector position. This can be easily obtained by correcting the forward kinematics using the vector of parameters ν computed in the previous Section and comparing the entire batch of measured 393 poses and the 343 poses not used for the identification with the ones expected.

The need for both validations rises considering the methodology that has been followed. The precision is expected to increase independently of the correctness of the model when considering any group of poses that includes the ones used for the identification phase, denoted as $\mathbf{\Psi}$. This is due to the least squared method, which produced the parameters that best approximate the position in those poses. On the other hand, when considering the remaining configurations, if the model has been configured correctly, the error must also decrease in the poses not used for the identification phase.

Tables 6.4 and 6.5 show the results obtained with this calibration process, respectively, considering the entire batch of 393 poses and only the 343 not containing $\mathbf{\Psi}$. Discarding the non-physical results obtained with options 1 and 3, the best performance is obtained by option 1_2, with an average increase of 80.7% in the mean accuracy, from 28.7 mm to 5.54 mm.

	Mean [mm]	Max [mm]	SD [mm]	Mean Reduction [%]
Nominal	28.7	57.3	12.3	-
Option 1	5.42	16.4	2.59	81.1
Option 2	16.9	36.4	6.05	41.0
Option 3	9.92	30.0	5.26	65.4
Option 4	6.50	16.4	3.46	77.3
Option 5	10.9	26.0	5.25	62.1
Option 1_2	5.54	15.6	2.49	80.7
Option 3_2	9.80	29.5	5.26	65.9

Table 6.4: Table presenting the first validation results, conducted on both the poses chosen for the identification phase and the remaining 343. The focus is on the mean error, maximum error, Standard Deviation (SD) of the error, and the mean reduction compared to the nominal case. Although the maximum decrease for the maximum error is obtained with Option 1, the best physically reasonable solution is option 1_2, with a 80.7% reduction in the mean error with respect to the uncorrected values.

	Mean [mm]	Max [mm]	SD [mm]	Mean Reduction [%]
Nominal	28.7	57.3	12.3	-
Option 1	5.51	16.4	2.64	80.8
Option 2	17.1	36.4	6.12	40.4
Option 3	9.80	30.0	5.38	65.8
Option 4	6.78	16.4	3.52	76.4
Option 5	11.0	26.0	5.29	61.8
Option 1_2	5.64	15.6	2.55	80.3
Option 3_2	9.70	29.5	5.39	66.2

Table 6.5: Table presenting the first validation results, conducted only on the poses not chosen for the identification phase. The focus is on the mean error, maximum error, Standard Deviation (SD) of the error, and the mean reduction compared to the nominal case. It is interesting to notice how the mean error reduction is lower than the one of Table 6.4 as expected, however of only less than 1%. This means the parametrization of the model works effectively also on the poses not used for the identification phase.

To facilitate the comparison of the results between the different options, the mean error and the standard deviation from Tables 6.4 and 6.5 can be used to generate a Gaussian distribution of the error for each option. The corresponding results are shown in Figure 6.4 for Table 6.4, which includes all 393 poses, and in Figure 6.5 for Table 6.5, which excludes the 343 poses containing $\boldsymbol{\Psi}$.

The black line represents the nominal error, serving as a reference. Each calibration option adjusts the Gaussian curve, shifting it closer to the origin, while the peaks of the curves become more pronounced. This suggests that the error distribution is increasingly concentrated around smaller values, indicating improved accuracy.

It is important to note that the purely elastic model proves more accurate than the purely kinematic one. Moreover, Option 5, which is a variation of Option 3 without the parameter k_0, closely matches the performance of Option 3. In contrast, Option 4, derived from Option 1 by removing k_0, deviates significantly more from Option 1. The difference is approximately 4% for Options 3 and 5, while for Options 1 and 4, it reaches around 14%. This indicates that when considering both elasto-kinematic parameters, the elasticity of the base structure supporting the arm has a more substantial influence than when accounting solely for elasticity.

A noteworthy observation arises when comparing Option 1 and Option 3 with their corresponding physically feasible versions, Option 1_2 and Option 3_2. The adjustments made in Options 1_2 and 3_2 account for physical constraints and provide a more realistic solution, nevertheless they sacrifice less than 1% in accuracy.

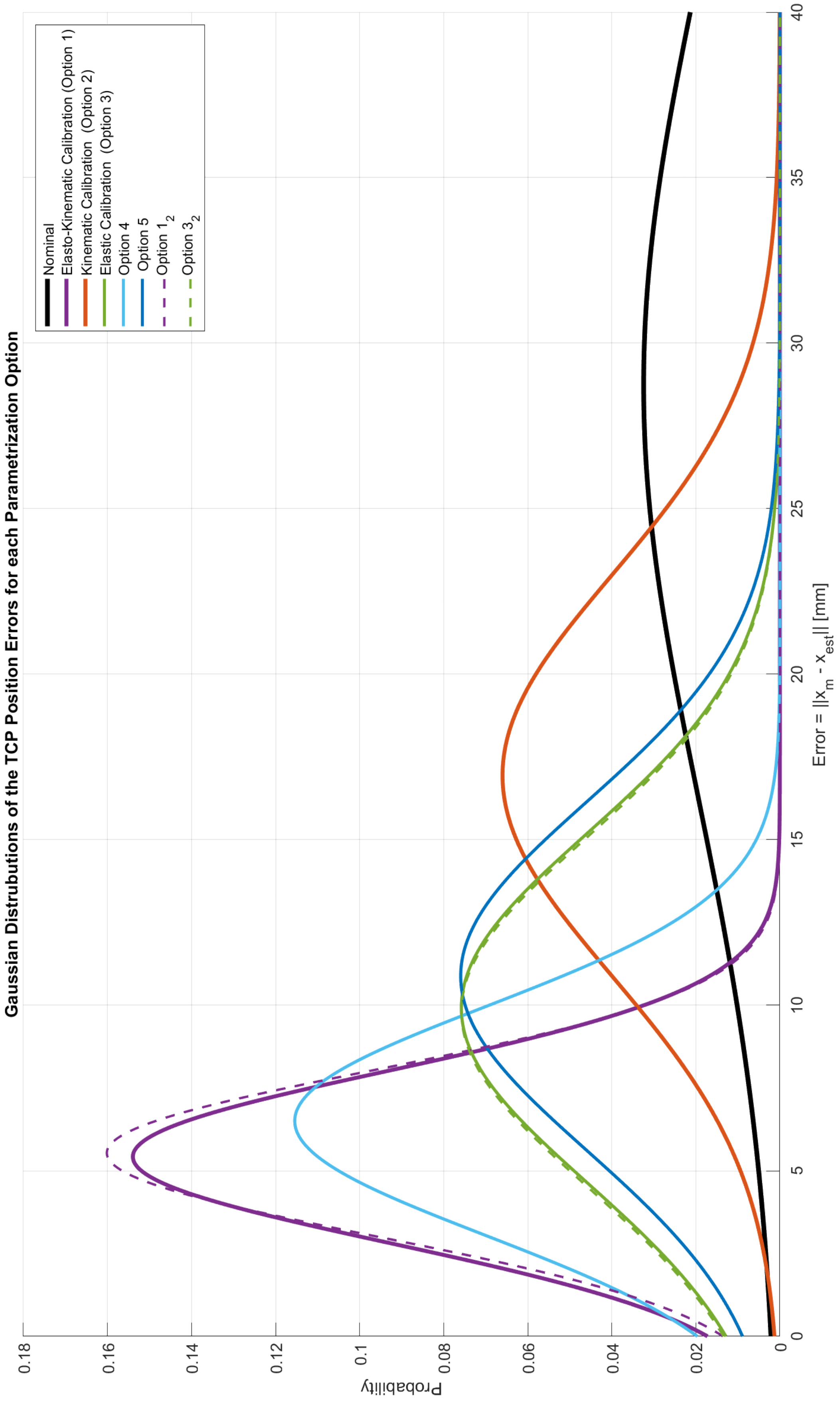


Figure 6.4: Plot of the error distribution for each different option when considering both the poses chosen for the identification phase and the remaining 343.

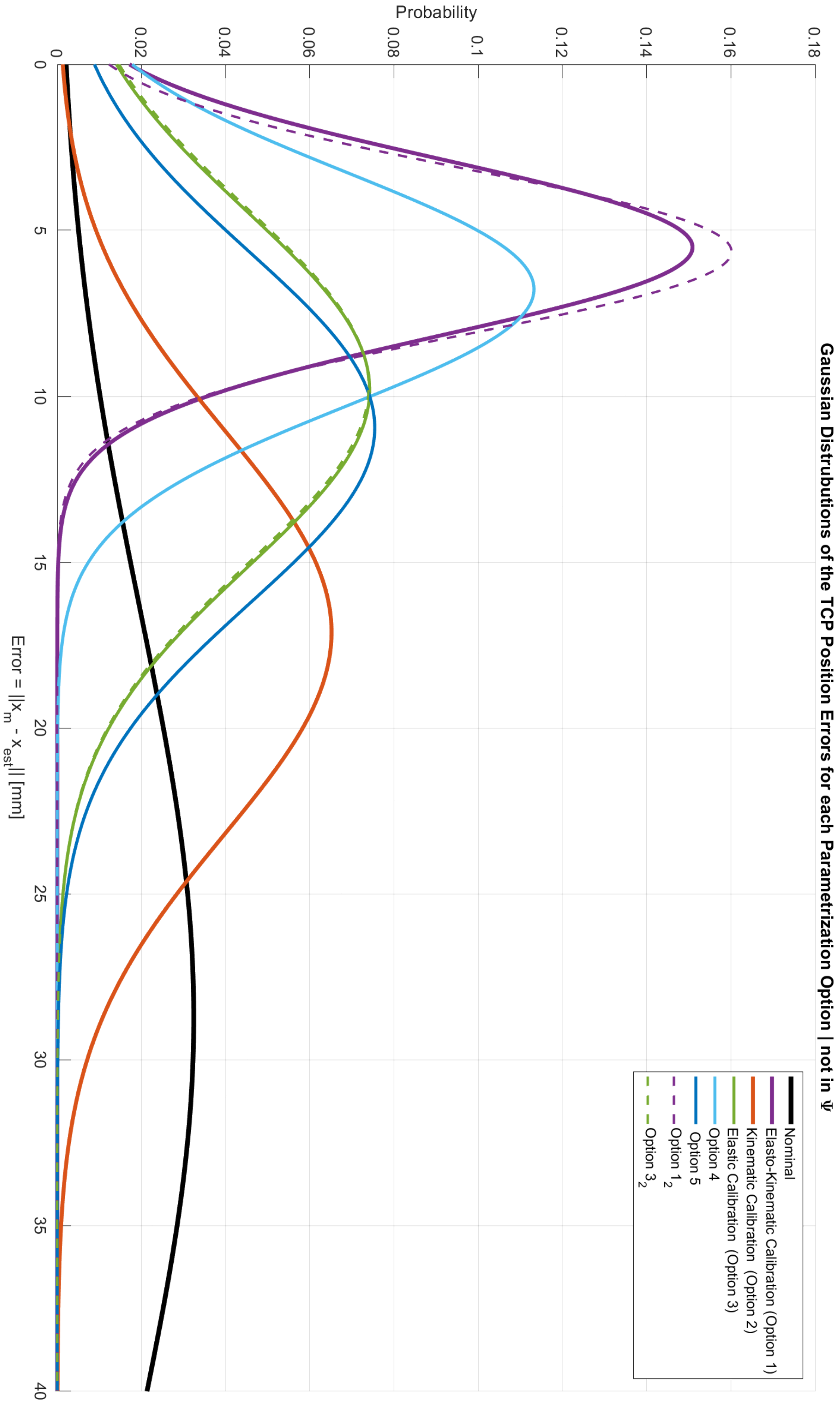


Figure 6.5: Plot of the error distribution for each different option when considering only the poses not chosen for the identification phase.

Although the behaviour of the joints' elasticity is expected to be linear, as discussed in Chapter 5.3, it was essential to validate this theory more thoroughly. This validation was conducted by repeating the calibration process with an increased load applied to the arm's end-effector, followed by comparing the results with those obtained previously under no load. If the theory holds true, the optimal set of parameters identified during this process should also be effective when the arm is unloaded, and vice versa.

The following tables 6.6 and 6.7 display the results obtained when applying the coefficients retrieved from the measurements with a load of 66.3 g applied at the end-effector. As expected, the first option, the elastic and kinematic calibration, proves again to be the best parametrization, reducing the end-effector position error of 82.8%, from an average of 32.9 mm to 5.65 mm. The average of 32.9 mm, which is higher than the previous no-load average of 28.7 mm, clearly indicates that the joints' elasticity significantly impacts the end-effector's position.
As previously mentioned, only five options are available in contrast to the unloaded case. The identification results were already physically acceptable, meaning all the elastic coefficients were positive, and therefore, the application of constraints was not required.
It is interesting to note that, once again, the best option is Option 1, which achieves a mean reduction of over 80%. When comparing the results obtained from all 393 poses with those derived from only 343 poses, excluding $\boldsymbol{\Psi}$, the difference is 0.4%. This finding mirrors the results observed when no load was applied to the end-effector, reinforcing the conclusion that the 50 poses used for identification, which constitute $\boldsymbol{\Psi}$, effectively lead to parameters that accurately describe the behaviour of the arm across the remaining 343 poses.

	Mean [mm]	**Max [mm]**	**SD [mm]**	**Mean Reduction [%]**
Nominal	32.9	66.3	13.8	-
Option 1	5.65	14.1	2.65	82.8
Option 2	19.0	38.4	6.81	42.1
Option 3	10.0	30.1	5.49	69.5
Option 4	6.97	18.3	4.01	78.8
Option 5	11.4	28.1	5.77	65.5

Table 6.6: Table displaying the validation results conducted applying the parameters obtained through the measurements with a load of 66.3 g applied at the end-effector, to the forward kinematics of the arm. The focus is on the mean error, maximum error, Standard Deviation (SD) of the error, and the mean reduction compared to the nominal case. These values are retrieved considering all the 393 poses measured.

	Mean [mm]	Max [mm]	SD [mm]	Mean Reduction [%]
Nominal	32.9	66.3	13.8	-
Option 1	5.77	14.1	2.72	82.4
Option 2	19.2	38.4	6.92	42.5
Option 3	9.95	30.1	5.60	69.8
Option 4	7.26	18.3	4.09	77.9
Option 5	11.5	28.1	5.86	65.0

Table 6.7: Table displaying the validation results conducted applying the parameters obtained through the measurements with a load of 66.3 g applied at the end-effector, to the forward kinematics of the arm. The focus is on the mean error, maximum error, Standard Deviation (SD) of the error, and the mean reduction compared to the nominal case. These values are retrieved considering only the 343 poses not used for the validation phase.

The following pages present a similar analysis conducted on the unloaded results, specifically focusing on the Gaussian distribution of the errors. Figures 6.6 and 6.7 compare the outcomes of the five different calibration options derived from the measurements of the loaded arm.
As seen in Figures 6.4 and 6.5, the first option, which employs elasto-kinematic calibration, clearly stands out as the most accurate among the alternatives. The Gaussian curve for this option is positioned closest to the origin, indicating a lower mean error and a more concentrated error distribution.
Moreover, the trends observed for the remaining options are notably similar, suggesting that they perform comparably under the unloaded conditions. This consistency in results reinforces the reliability of the elasto-kinematic approach across various testing scenarios.

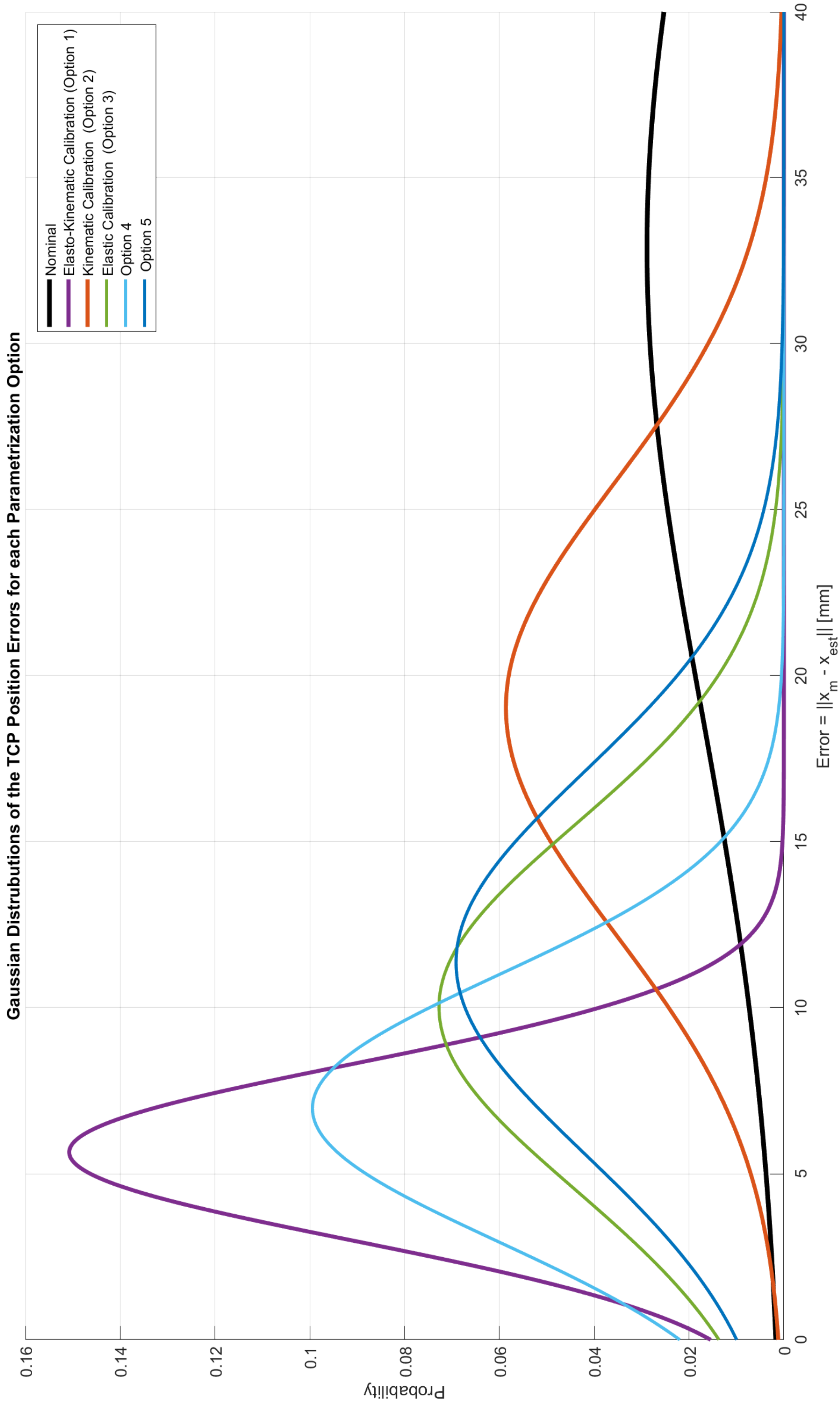


Figure 6.6: Plot of the error distribution for each different option when considering both the poses chosen for the identification phase and the remaining 343, using a load of 66.3 g at the end-effector.

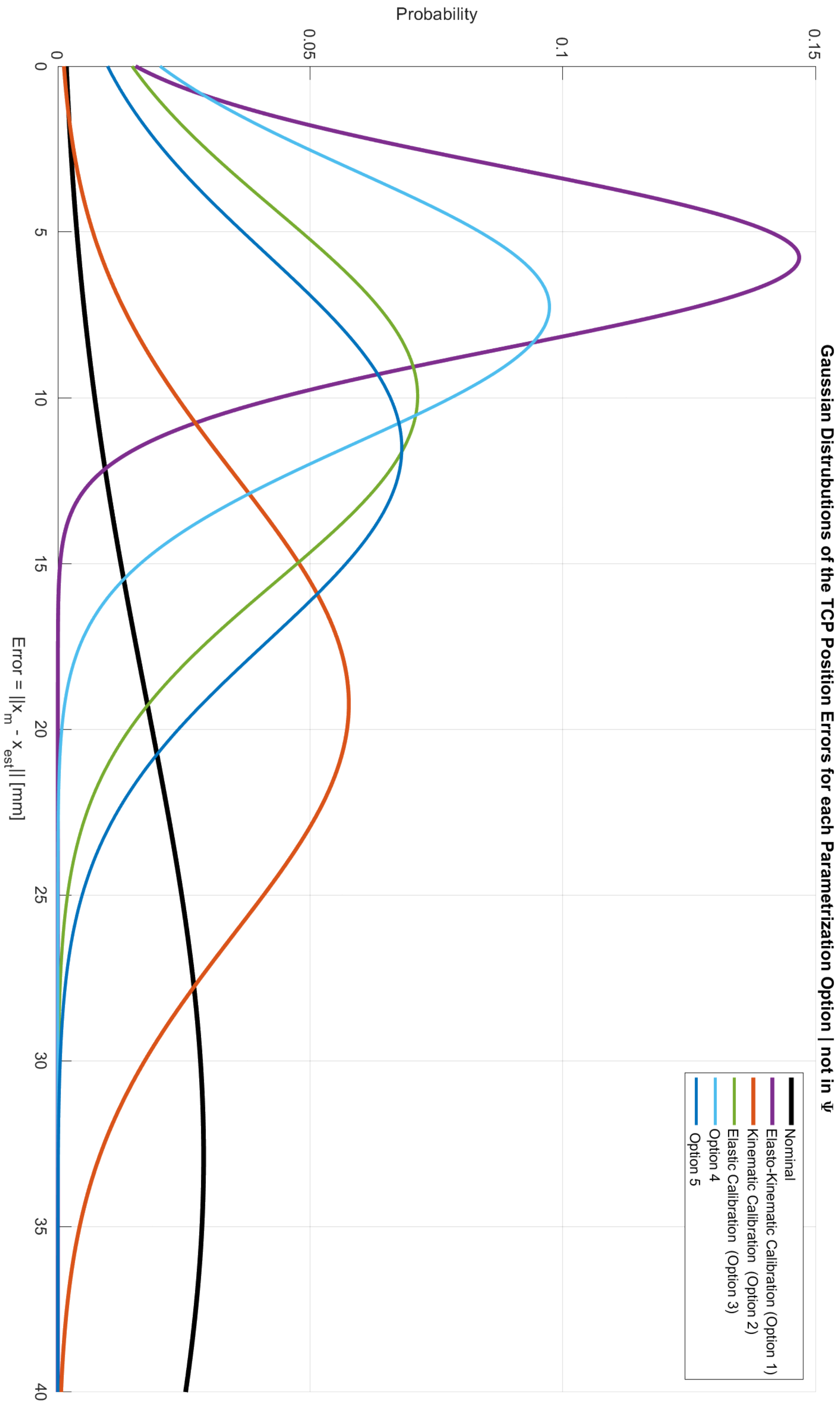


Figure 6.7: Plot of the error distribution for each different option when considering only the 343 poses not chosen for the identification phase, using a load of 66.3 g at the end-effector.

The final step of the validation process involves a critical comparison of how the sets of parameters identified using two distinct sets of poses perform under opposite conditions. Specifically, one set is derived from measurements taken with no load applied at the end-effector, while the other set is based on measurements taken with a load of 66.3 grams. This comparison is essential for ascertaining the reliability of the identified parameters and determining whether they can be effectively applied across different loading conditions. If the assumption of linearity in joint elasticity holds true, it is anticipated that both parameter sets should function adequately when tested in the alternate scenario.

This implies that the performance of the parameter set $\boldsymbol{v}_{12}$, obtained applying the set of parameters of option 1_2 to the measurements with no load at the end effector, will be evaluated using the loaded pose set; the parameter set $\boldsymbol{v}_{1H}$, obtained applying the set of parameters of option 1 to the measurements with a 66.3 g load at the end effector will be applied to the unloaded pose set. The results of this comparative analysis are systematically presented in Table 6.8, and a further detailed illustration of the position errors is illustrated in histograms 6.8 and 6.9.

Interestingly, despite the fact that $\boldsymbol{v}_{1H}$ was not identified using the unloaded pose set, it yields an average error of 5.50 mm. This result is slightly lower than the 5.64 mm error produced by the parameter set $\boldsymbol{v}_{12}$.

One possible explanation for this unexpected phenomenon is that the elastic errors introduced by the load at the end-effector may have partially concealed other types of errors. These could include errors arising from various factors, such as cabling issues, backlash in the joints, and other complexities not adequately accounted for in the original model. As a result, the parameter set $\boldsymbol{v}_{1H}$ emerges as the most suitable candidate for use in the integration phase of the project, which will be discussed in the following chapter.

	Error in poses with no load		**Error in poses with 66.3 g load**	
Parameters	**Mean [mm]**	**Max [mm]**	**Mean [mm]**	**Max [mm]**
$\boldsymbol{v}_{12}$	5.64	15.6	6.14	13.7
$\boldsymbol{v}_{1H}$	5.50	17.4	5.77	14.1

Table 6.8: Comparison of the results obtained using two different sets of parameters,$\boldsymbol{v}_{12}$ and $\boldsymbol{v}_{1H}$, in the no-load scenario and when applying 66.3 g at the end-effector.

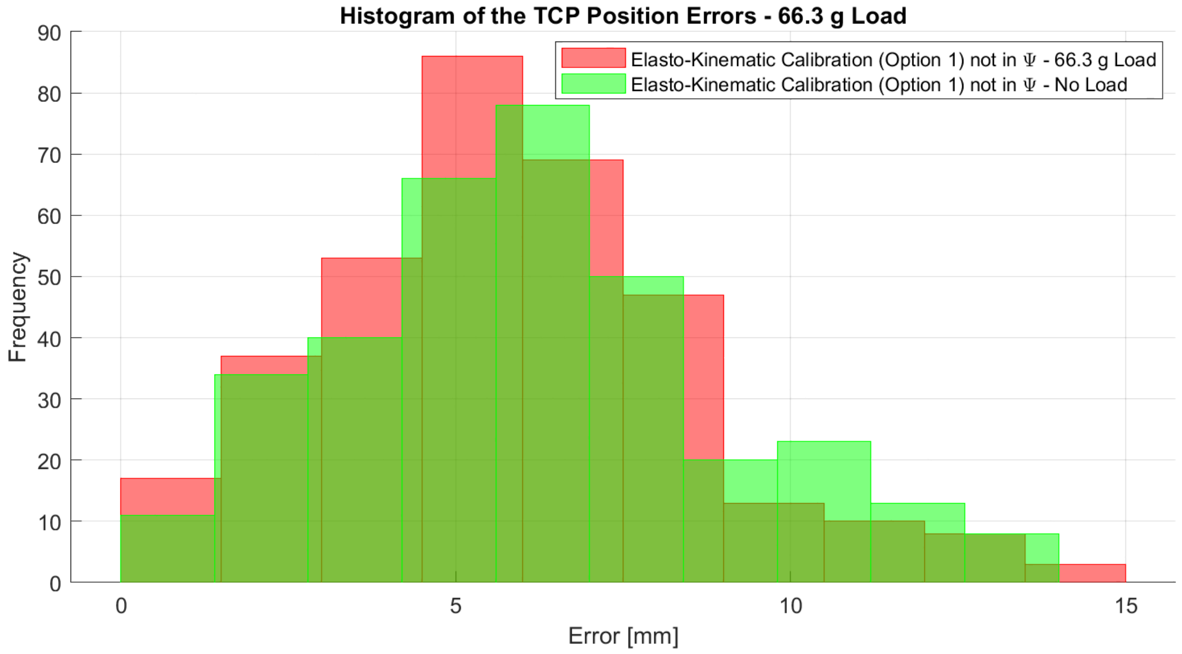


Figure 6.8: Histogram showing the distribution of the errors in the measurements using a 66.3 g load at the end-effector, comparing applying the correction $\mathbf{v}_{1H}$ obtained from the measurements with a 66.3 g load at the end effector, and $\mathbf{v}_{12}$ obtained from the measurements without load.

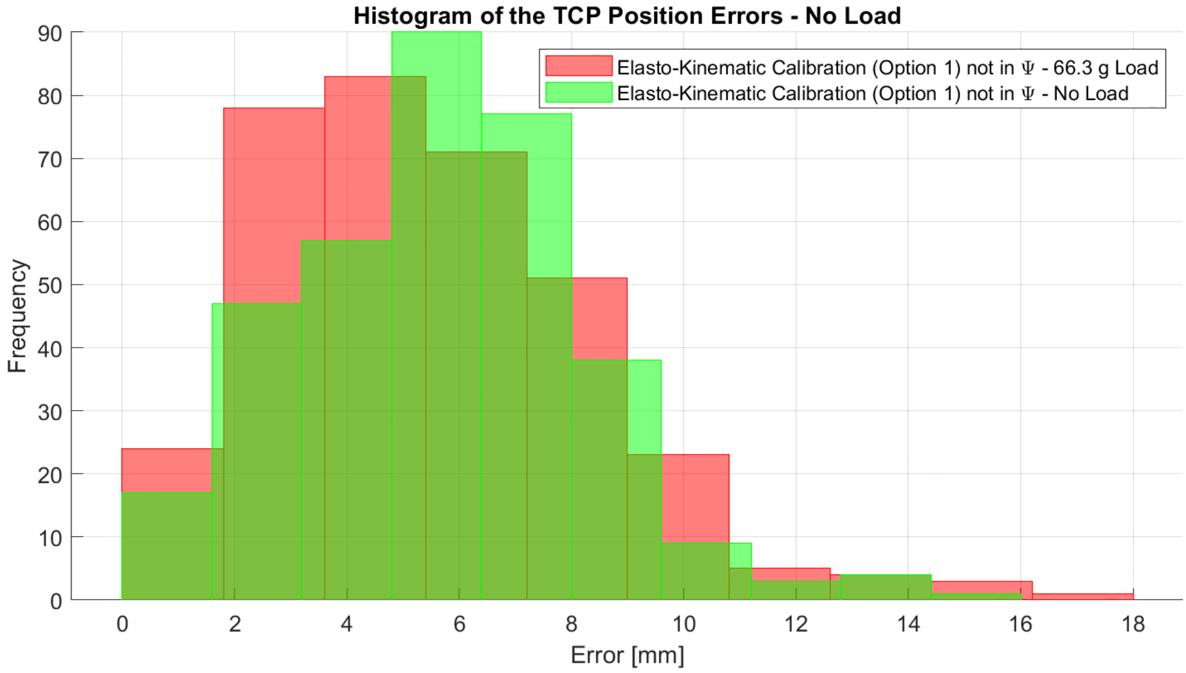


Figure 6.9: Histogram showing the distribution of the errors in the measurements using no load at the end-effector, comparing applying the correction $\mathbf{v}_{1H}$ obtained from the measurements with a 66.3 g load at the end effector, and $\mathbf{v}_{12}$ obtained from the measurements without load.

7 Integration

This chapter focuses on the final phase of the calibration process: integrating the corrected kinematics into the arm control software. This involves implementing the inverse kinematics library into the software architecture of LRM based on Links and Nodes and then adjusting the inverse kinematics to account for each link's inaccuracies and the joints' elasticity.

7.1. Links and Nodes

Already introduced in Chapter 1, Links and Nodes (Schmidt, 2023) is the in-house software developed at the Robotics and Mechatronics Institute of DLR for coordinating and integrating various processes within robotic systems. As this thesis focuses on advancing the development of the LRM to a state where the robotic arm is fully functional and the calibration is integrated into its software architecture, the final step involves incorporating the inverse kinematics into Links and Nodes and adjusting it to account for the calibrated parameters.
A visual representation of Links and Nodes is shown in Figure 7.1, where it is possible to identify the main processes needed to operate the arm:

- **rover_communication**
  Python process that enables the communication between the onboard PC and the PCB to which the arm servos are connected. It is responsible for converting angular data into PWM signals, as explained in Chapter 3. Once initiated, it waits for the 'LRM_simulink' to start, then monitors until the 'example_notebook' publishes to the 'Arm' topic, which is subscribed to by 'rover_communication'. The first level calibration of Chapter 3 is implemented in this process, as a simple interpolation before the PWM conversion.
  LRM has two main control modes: one for enabling locomotion using the wheels and the other for controlling the arm. The selection between these modes is determined by the first byte of the message sent from the PC to the PCB, which is hardcoded to '1' for arm control.

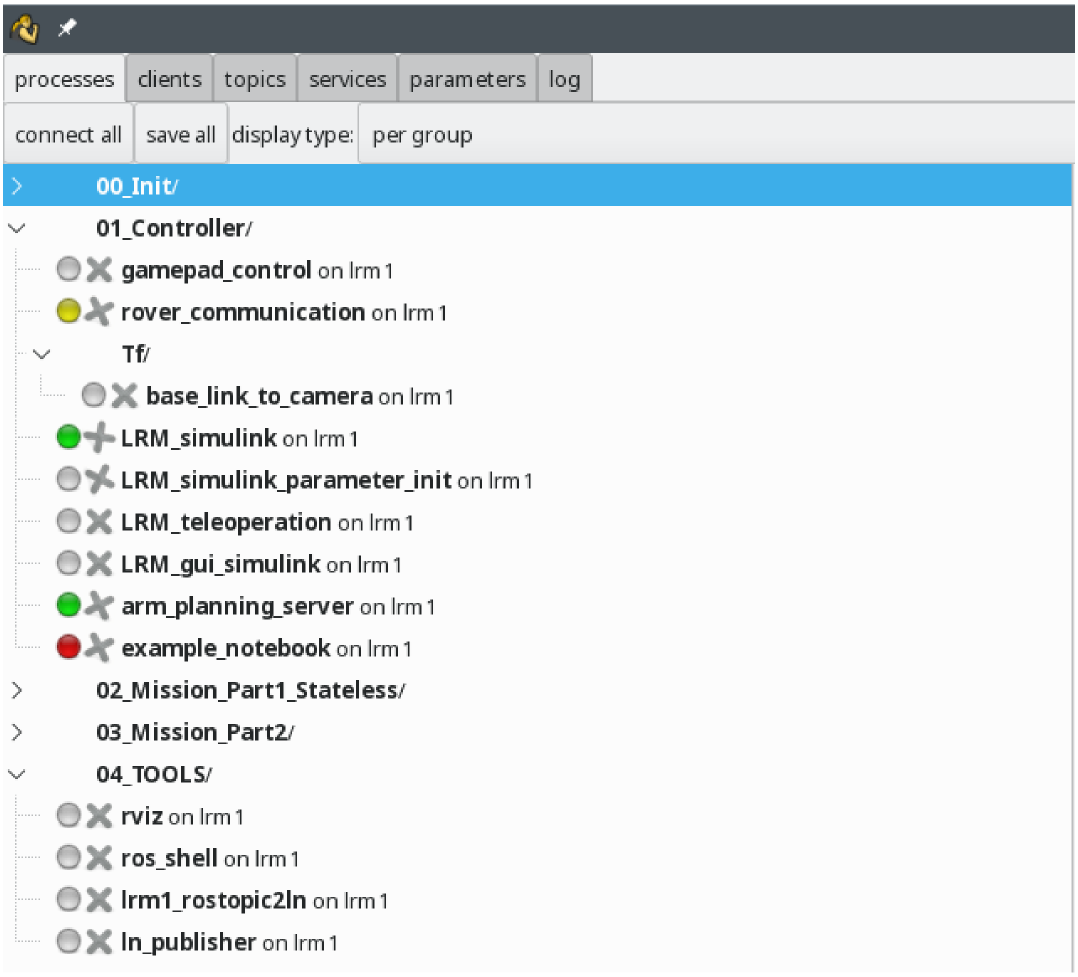
- **LRM_simulink**
  Simulink process manages the rover's overall kinematics. It receives the commands from the gamepad or the autonomous navigation stack as input and publishes the data to control the wheels to 'rover_communication'. It is essential to run the Simulink process, as without its activation, neither 'rover_communication' nor any other operations would start.

- **arm_planning_server**
  The software used to calculate the inverse kinematics path was developed internally at

DLR's institute and is a Python library called Robot Motion Planning Library (RMPL). Utilizing this continuously updated package can effectively leverage the advanced inverse kinematics, path generation, and collision avoidance features of the LRU system (Wedler et al., 2015).

- **example_notebook**
  A live script serves as the user interface for controlling the arm. It loads the URDF model and provides two different control modes: users can either control each joint individually and send them to the desired positions using simple forward kinematics, or specify a goal position and employ inverse kinematics to guide the arm while avoiding self-collisions. Although the RMPL library allows for direct subscription to the generated path on the Links and Nodes topic, the need to integrate elastic-kinematic corrections prompted the decision to have the notebook publish the data to LRM_communication instead.

In summary, once example_notebook creates the desired path importing the correct IK solver and other necessary functions from ”arm_planning_server”, the joint configuration for each waypoint is published to the topic ”Arm”. Then, the topic ”rover_communication”, which is subscribed to ”Arm”, sends the correct PWM signal to the servos.



```
> 00_Init/
v 01_Controller/
    gamepad_control on lrm1
    rover_communication on lrm1
  v Tf/
      base_link_to_camera on lrm1
    LRM_simulink on lrm1
    LRM_simulink_parameter_init on lrm1
    LRM_teleoperation on lrm1
    LRM_gui_simulink on lrm1
    arm_planning_server on lrm1
    example_notebook on lrm1
> 02_Mission_Part1_Stateless/
> 03_Mission_Part2/
v 04_TOOLS/
    rviz on lrm1
    ros_shell on lrm1
    lrm1_rostopic2ln on lrm1
    ln_publisher on lrm1
```

Figure 7.1: The Links and Nodes Manager Interface used to operate the Lunar Rover Mini. Each node is a process and can be activated individually.

7.2. Elastic and Kinematic Compensation

The final step is integrating the parameters obtained for the elastic and kinematic calibration into the inverse kinematics. As discussed in the previous chapter, the most effective set of parameters is $\boldsymbol{\nu}_{1H}$, obtained through the elasto-kinematic correction by measuring the arm with a load of 66.3 g at the end-effector.

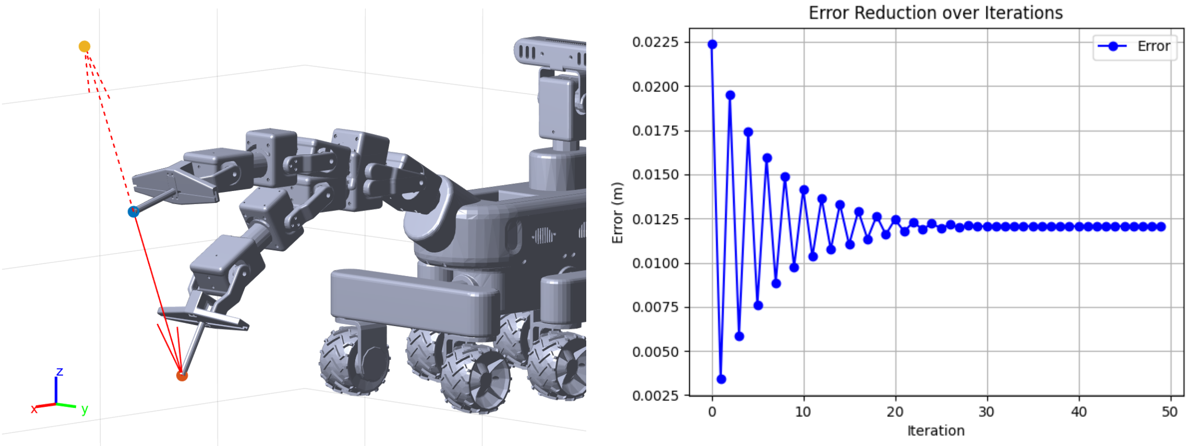


Figure 7.2: On the left, the first method approached to solve the inverse kinematics taking into account the calibration parameters, involving an external forward kinematics analyser. On the right, the result of this method, showing how the TCP position error for the goal position decreases with respect to the initial offset, however not converging to zero.

Once the parameters are loaded, an external script dealing with the corrected forward kinematics must be implemented. Given a determined pose, meaning the angle of each joint, the script can output the real end-effector's position corrected by the parameters.

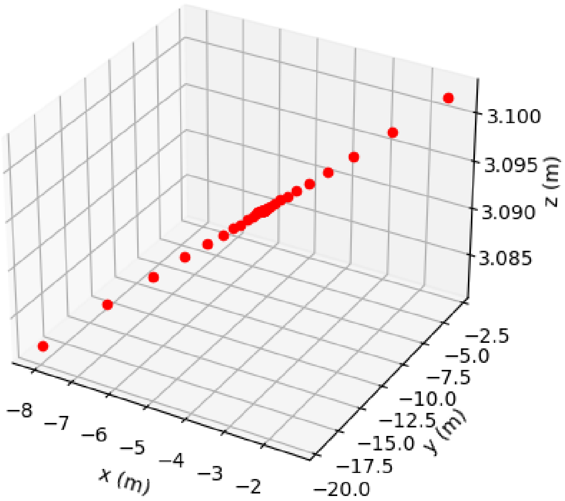


Figure 7.3: 3D Plot displaying the position of the TCP in each iteration of the first method used to solve the calibrated inverse kinematics.

The initial approach involved placing this script in a loop and feeding its output to the inverse kinematics. Using the error vector derived from the difference between the goal position and the actual position obtained from the corrected forward kinematics, a new temporary goal position would be identified. This process would continue until the actual position from the corrected forward kinematics is sufficiently close to the desired position. This process is visually represented on the left side of Figure 7.2. In the figure, the blue dot represents the desired end-effector position, while the red dot shows where the end-effector would end up due to elasticity and kinematic offsets (greatly exaggerated and not to scale with the actual deflections). The red vector represents the error vector, and the red dotted vector, which points toward the yellow dot, is the inverse of the error vector. This yellow dot is the first approximation of the new goal for the inverse kinematics, allowing the corrected forward kinematics to output a position closer to the original goal (blue dot). A flowchart representing this process is shown in Figure 7.4.

Testing this solution with different goal positions led to unexpected and unsatisfactory results. The error does not reduce to zero but stabilizes at an unacceptable value of approximately 1.2 cm. This is likely due to the algorithm converging to a local minimum, where the inverse kinematics corrections cannot fully compensate for the error. The plot on the right of Figure 7.2 shows how the error stabilizes at a value well above the expected zero, and Figure 7.3 shows a 3D representation of the goal position in each iteration. From this plot, it is clear how the algorithm forces the goal position to be in a straight line, ignoring the possibility of exploring other directions.

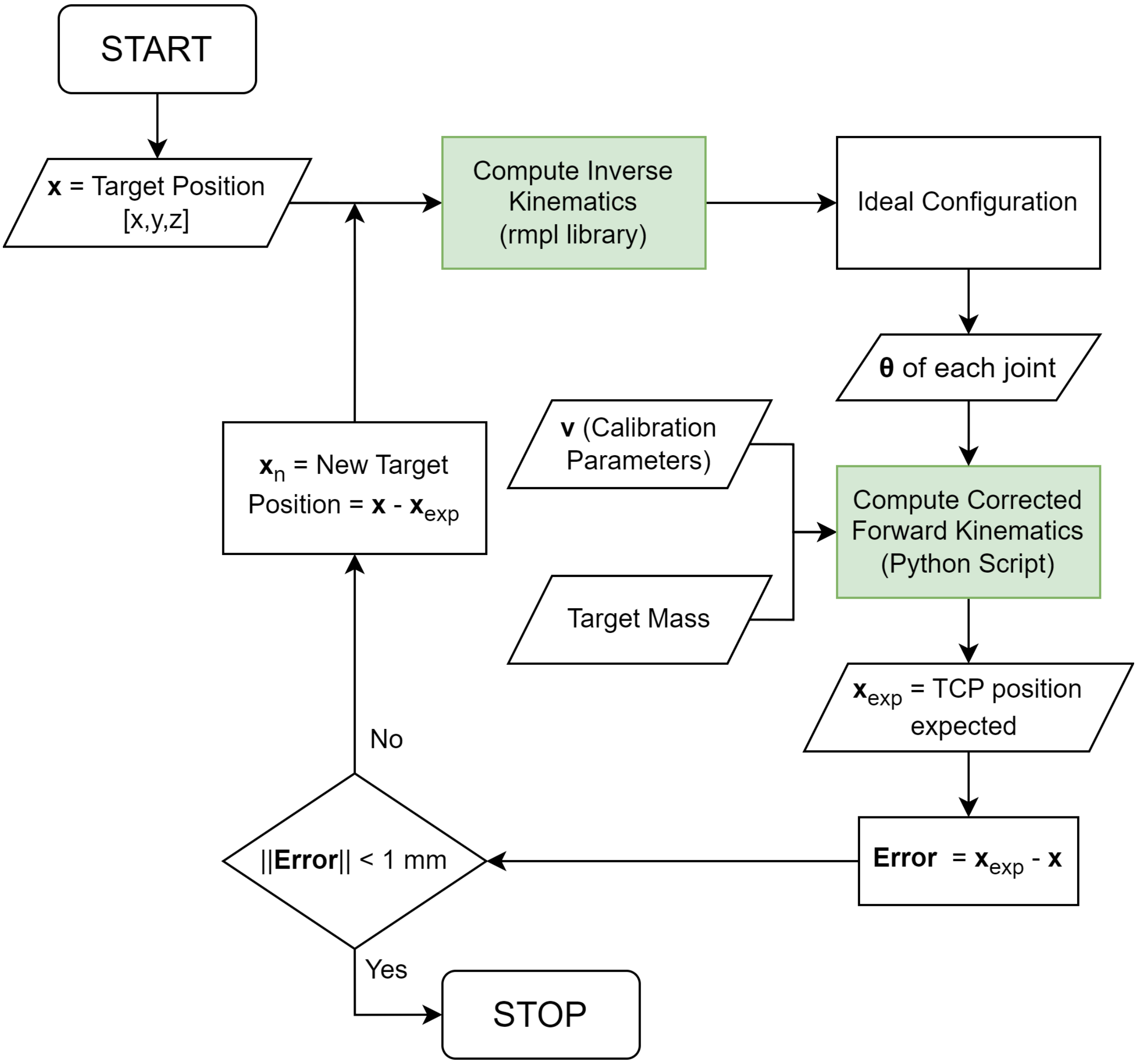

START
x = Target Position [x,y,z]
Compute Inverse Kinematics (rmpl library)
Ideal Configuration
θ of each joint
v (Calibration Parameters)
Target Mass
Compute Corrected Forward Kinematics (Python Script)
x_exp = TCP position expected
Error = x_exp - x
||Error|| < 1 mm
No
x_n = New Target Position = x - x_exp
Yes
STOP

Figure 7.4: Flowchart of the first method used to correct the inverse kinematics with the calibration parameters. Using as inputs the mass of the load acting on the TCP, as well as the correction parameters, the process involves a while loop that alternates the use of IK and FK to converge to the goal position.

A straightforward solution to this issue involves leveraging the higher precision observed during the first iteration of the loop. As shown in the plot in Figure 7.4, the first iteration, marked by the orange dot on the left, reduces the error significantly from around 2 cm to approximately 3 mm. This suggests that the final solution likely lies near this position. Rather than continuing the loop indefinitely, it may be sufficient to explore this neighbourhood by testing different goal positions within a defined 1x1x1 cm cube centred around the orange dot. A graphic interpretation of this process is presented in Figure 7.5, and a flowchart illustrating this approach is displayed in Figure 7.6.
When this approach was empirically tested in different scenarios, using different goal positions, the error consistently dropped below the threshold of 1 mm.

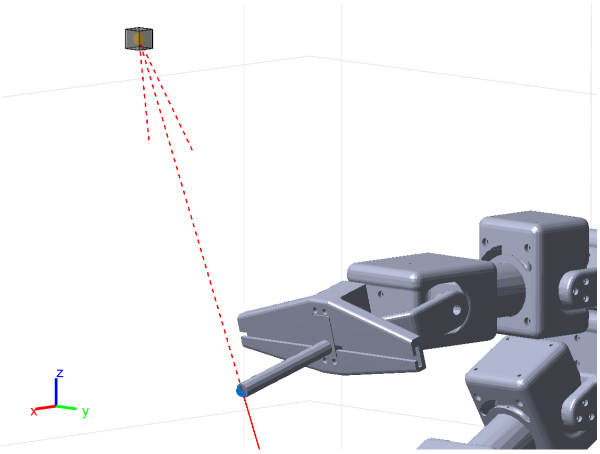


Figure 7.5: Visual representation of the cubic volume surrounding the initial iteration result, used in the second method for solving the corrected IK.

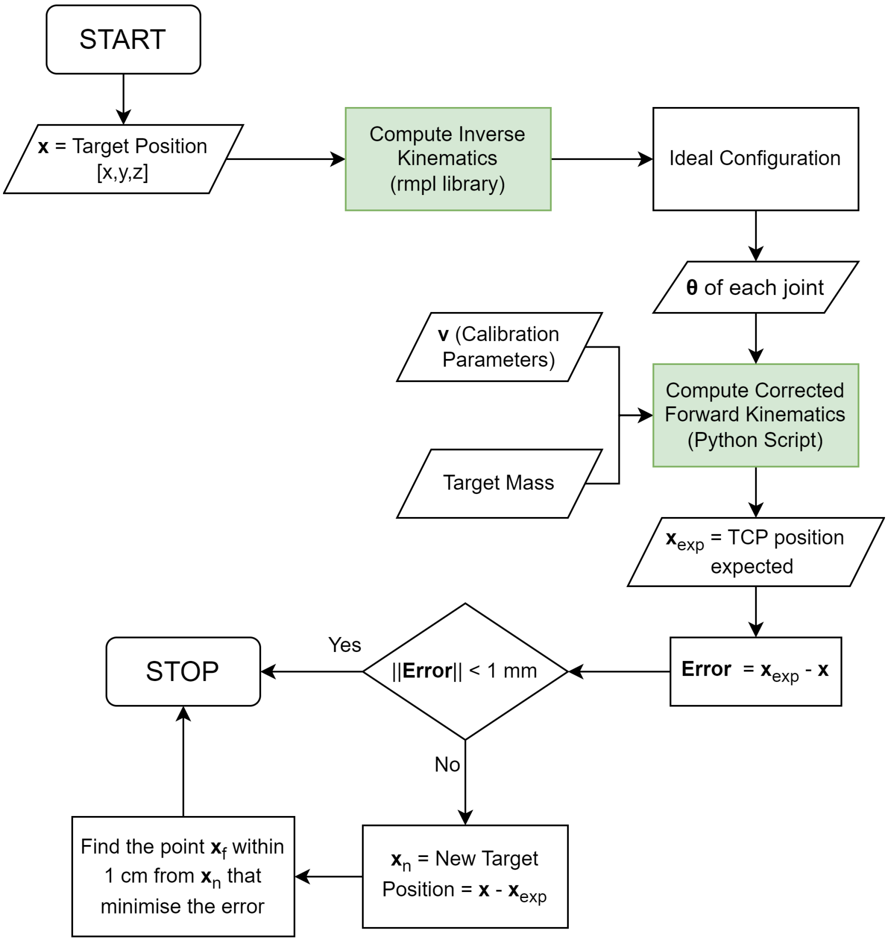

START
x = Target Position [x,y,z]
Compute Inverse Kinematics (rmpl library)
Ideal Configuration
θ of each joint
v (Calibration Parameters)
Target Mass
Compute Corrected Forward Kinematics (Python Script)
x_exp = TCP position expected
Error = x_exp - x
||Error|| < 1 mm
Yes
No
STOP
x_n = New Target Position = x - x_exp
Find the point x_f within 1 cm from x_n that minimise the error

Figure 7.6: Flowchart of the second method approached to solve the IK considering the calibration parameters, involving an external FK analyser. In this case, instead of re-entering the loop feeding the IK with the data of the corrected FK, the vicinities of the new goal position are scouted.

8

Conclusion and Future Recommendations

This conclusive chapter will summarise the methodology used to answer the research questions, focusing on the results obtained for each of them, followed by some recommendations for future studies on the same topic.

This thesis has tackled the significant challenge of improving the accuracy and operational reliability of the 6 DOF robotic arm developed for the Lunar Rover Mini project, conducted within the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). Driven by the general ideology of the LRM project, which consists of creating an advanced but cost-effective rover to be used as a learning platform, the employment of 3D-printed plastics and inexpensive RC servos presented an obstacle to the goal of integrating a precise manipulator. RC servos are notoriously inconsistent and prone to offsets which, when compounded across the 6 degrees of freedom of the LRM robotic arm, result in substantial inaccuracies in the TCP's position. Additionally, these servos are not designed to handle sustained loads and exhibit common motor issues, such as backlash, in a much more pronounced manner. On top of that, using ASA plastic and assembling the arm without any metal parts inevitably leads to noticeable offsets that worsen the arm's positioning accuracy. These structural imperfections, when combined with the inherent limitations of the RC servos, significantly hinder the arm's ability to maintain precision, especially under load, which is the goal function of the arm: being able to grab samples from the ground and release them in the small basket on its top avoiding collisions with the rest of the rover.

The peculiarity of this system, which forced the choice between the model-based and model-less calibration towards the former option, is the absence of torque feedback sensors in its motors. For this reason, a model-less calibration is not possible, and therefore this thesis focussed on the general steps used to follow a model-based approach (Kamali et al., 2016): first, the arm kinematics are modelled, as well as the parameters that may induce inaccuracies in the position of the TCP; then the end-effector is measured in different known poses, which are then compared to the ones obtained through the forward kinematics. This comparison is finally implemented in a least-squared algorithm to identify the parameters, which are then integrated into the inverse kinematics software controlling the arm.

8.1. Results Summary

Following the premises of the introduction to this final chapter, it is hereby reported the original research question:

How can a 6 DOFs 3D-printed robotic arm built with inexpensive motors improve its accuracy through a model-based joint deflection compensation?

The answer to this question lies in the concept followed by this work, the 3rd level calibration, or elastic and calibration, tailored by the kinematic calibration. As explored in Chapter 2, current literature supports this as the most effective approach for addressing the issue. However, none of the studies reviewed dealt with a system as imprecise as this one, so extra caution and attention to detail were required throughout the entire project.

First, a deep analysis of the offsets of each servo implemented in the arm was necessary to cover the first-level calibration. This step was fundamental to ensuring the commands from any software commanding the arm would translate into the correct angles. Using a self-designed device similar to a protractor, each servo was tuned to be perfectly aligned to the 0° position, as well as the +100° and -100°. This choice was not arbitrary but was guided by the next steps of the calibration, which involve measuring the arm with the joints at exactly those three angles.
Then, preliminary tests had to be conducted to ensure the methodology would fit the requirements, and answer the first subquestion **SUBQ-1**, here reported: *Considering the limitations of inexpensive RC servo motors, what model-based approach can effectively compensate for elastic deflection within the joints under varying load conditions?*.
Testing the deflection of a single servo under varying loads yields a clear response to this question. Figure 5.7 shows the trend of the angular deflection for one of the servos used in the first three joints of the arm. The result is an almost perfectly linear behaviour, which, along with the fact that the non-elastic parameters represent simple kinematic offsets expected to be independent of the pose of the arm, supports the use of a simple least-squared algorithm to identify the calibration parameters.

This algorithm requires at least one equation for each unknown variable to develop a solution. Consequently, the number of poses that need to be measured depends on whether both the position and orientation of the arm's TCP are measured, or only the position. Although also considering the orientation would drop the number of required measurements by half, as the equations for each measurement would be 6 (x, y, z, α, β,γ), the higher precision in the position of the markers used for the ART Track lead to the choice of using only the position, with the advantage of also simplifying the post-processing analysis and the parameter identification process. With this premise, it is possible to address the following subquestion **SUBQ-2**: *To what extent should the model developed for joint deflection compensation be simplified, considering the trade-off between accuracy and computational complexity suitable for the intended control system of the robotic arm?*
The complexity of the calibration process was further reduced by not considering the lateral elastic deformations of each joint. This decision was guided by the exponential increase of complexity such a feature would have led to when considering the compensation of the elasticity in the inverse kinematics problem. Also, referring again to 5.7, a value of $3 \cdot 10^{-2}$ rad/Nm for the angular elastic coefficient of the joints indicates the vast majority of the deformation will be caused by them, and not by the links, as often mentioned in literature (De Luca & Book, 2008).

SUBQ-3 states: *How should the common issues of target occultation and collision avoidance between the arm and its surroundings during measurement be addressed?* To address this subquestion, it is necessary to refer to Chapter 5.2. Before starting the measurement phase, it was necessary to analyze the workspace around the arm to ensure that none of the chosen poses would result in self-intersections or environmental collisions. Of the $3^6 = 729$ possible pose combinations, only 393 were deemed reliable and feasible based on the minimum distance between the end-effector and the base structure, as well as the other links.

The next subquestion, **SUBQ-4**, poses: *To what extent does joint elasticity influence the position of the end effector, compared to kinematic offsets caused by mechanical defects?*
To answer this subquestion, Figures 6.6 and 6.7 are useful, as they show the results of the validation conducted with the remaining 393 poses not used for the parameter identification phase. The five parametrization options are compared, including the elasto-kinematic calibration, the purely elastic and purely kinematic cases, and the elasto-kinematic and elastic cases where the elasticity of the base structure supporting the entire weight of the arm is not modelled. Comparing them, from the Gaussian plots of the TCP position error it is clear how the elasto-kinematic calibration is the most efficient one, however insightful considerations can be made regarding the other curves; The purely kinematic calibration is insufficient to cover the majority of the errors that lead to inaccuracies in the TCP, while the purely behaviour is more promising. This result is clearer in Table 6.7, where it is stated that from a mean error of more than 3 cm, it drops to respectively around 2 cm and 1 cm using the purely kinematic and elastic set of parameters. This result highlights the significant influence joint elasticity has on the final position of the TCP, emphasizing the need to account for it and confirming the hypothesis presented in this thesis. On a side note, it is interesting to analyze the results of Options 4 and 5 while neglecting the flexibility of the base structure (k_0). Even without considering it, the results still outperform the purely kinematic calibration. Additionally, the 20% difference in the mean error between the elasto-kinematic correction and the corresponding option without k_0 confirms the hypothesis of the base structure flexing under the weight of the arm.

Furthermore, the arm was tested with two different loads at the end-effector. In one case, the only load was the target, while in the second case, an additional 66.3 g of aluminium disks were added. This approach aimed to confirm the hypothesis of linearity in the servo elasticity, as if the set of parameters retrieved from one case applies to the other, and vice versa, this would support the hypothesis. As visible in the histograms of Figures 6.9 and 6.8, the results of this validation flawlessly address **SUBQ-5**, which states: *What experimental validation methods can be used to assess the effectiveness of the gravity compensation strategy implemented, evaluating its impact on arm performance in various load scenarios?*. The histograms overlap almost perfectly, indicating that both sets of parameters can be used independently of the target mass. However, as shown in Table 6.8, the parameters obtained with the extra load are more accurate, with an average error of respectively 5.50 mm and 5.64 mm in the measurements with no load and 5.77 and 6.14 mm in the set with the 66.3 g load. This may result from the additional deformation during this set of measurements, which likely masked other non-modelled errors.

The last subquestion, **SUBQ-6**, reads: *How can the proposed model-based calibration and gravity compensation be effectively integrated into the control system of the robotic arm?*. To achieve this goal, the inverse kinematics library developed at DLR needed to be integrated into the Links and Nodes software used by the rovers to connect the various processes required for their operation. Then, the elasto-kinematic correction had to be implemented in the inverse kinematics, and two leading solutions were found. The first option involves running the inverse kinematics algorithm and then feeding its solution into a forward kinematics script that predicts the position of the TCP when the load is applied at a specific joint configuration. The error between the FK script output and the goal position is then calculated. If the norm of this error exceeds a certain threshold, it is subtracted from the goal position, and the process is repeated. Given that this solution frequently encountered local minimum problems and that the first iteration typically produced the closest approximation to the actual goal position, an alternative approach was explored. Starting from the result of the first iteration, a volume of 1x1x1 cm around the newly defined goal position is examined, and the best solution within this volume is selected.

Overall, the outcome of this work is an improvement of over 80% in the accuracy of the end-effector position (Table 6.6), along with the successful integration of the robotic arm into the Lunar Rover Mini system.
The complete open-source code is available at: https://github.com/MarcoConenna01/Thesis.

8.2. Future Work Recommendations

In light of the limitations encountered during this work and the potential benefits of having a highly precise manipulator on the Lunar Rover Mini, several avenues for future research and development can be proposed:

- **Improvement of Calibration Method**
  Although this thesis significantly increased accuracy, further refinement of the calibration process could yield even better results, such as compensating for the end-effector's orientation.

- **Enhanced Modeling of Elasticity and Deformations:**
  While joint elasticity was a key factor considered in this work, additional modelling of other sources of deformation, such as thermal expansion, external forces acting on the arm, or the motor backlash, could be valuable. These factors may further influence the precision of the end-effector, and accounting for them would enhance the overall robustness of the system.

- **Integration of Reinforcement Learning**
  Future work could explore the application of reinforcement learning to optimise the control of the manipulator. This would allow the system to learn from interactions with its environment and improve its performance over time, compensating for non-modelled errors and adapting to new tasks or conditions without manual recalibration.

- **Software and System optimisation**
  Optimising the Links and Nodes software to improve real-time communication and coordination between the manipulator and other rover systems could increase overall efficiency and responsiveness. Streamlining the code and enhancing the algorithms involved could also reduce the processing time for complex movements.

- **Real-time Environment and Target Object Detection**
  The Lunar Rover Mini already generates a 3D map of its surroundings for autonomous navigation. This map could enhance the inverse kinematics software by helping it avoid obstacles during the arm's movement toward a target and precisely locate the target itself. By integrating real-time environmental detection into the IK process, the manipulator could automatically plan and adjust its trajectory to avoid collisions with nearby objects, thereby making its operation more efficient and safer in complex environments.

- **Transition Dynamics and Damping Analysis**
  Analyzing the transition from one pose to another could yield valuable information on the system's damping characteristics. Developing a dynamic model of the arm that predicts oscillations based on a spring-damper-mass system would allow for better control of the manipulator's movements and improve accuracy during transitions between poses.

- **Implementation of Camera at the TCP**
  Adding a camera at the tip of the robotic arm's end-effector could significantly enhance the system's ability to perform delicate tasks requiring precise visual feedback. The camera could enable real-time monitoring of the arm's actions and assist in object detection. This setup would also be valuable for remote operations, allowing operators to visually verify the manipulator's actions, especially in tasks such as sample collection or tool manipulation in complex environments.

- **Implementation of RAFCON (DLR, 2024)**
  Implementing RAFCON, the state machine execution software developed by DLR, would provide a significant advantage in automating complex workflows for target detection and selection. RAFCON allows for the efficient orchestration of tasks, enabling the integration of visual detection systems and manipulators, thereby improving automation and target accuracy.

- **Active Stabilization System**
  Implement an active stabilization system where the arm can adjust its position dynamically to maintain the rover's centre of gravity. Sensors could be integrated to detect tilting angles, allowing the arm to extend or retract accordingly to stabilize the rover. Figure 8.1 illustrates a scenario where the rover has almost tilted over due to an encounter with a steep surface.

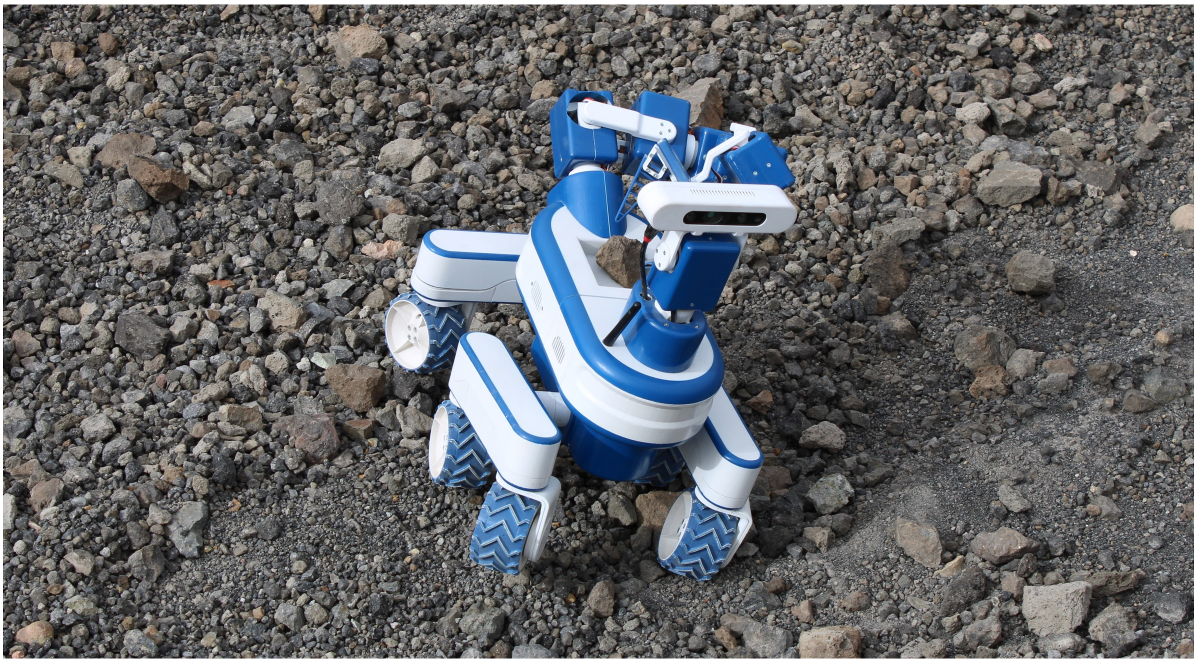


Figure 8.1: The Lunar Rover Mini navigating the rough terrain of Vulcano, in a precarious situation close to tipping over. In this scenario, utilizing the arm to shift the center of mass of the rover or leveraging it against the terrain could aid in recovering the rover to an upright position.

Bibliography

AGFRC. (2024). AGF: High torque digital servos. https://www.agf-rc.com/

Balanji, H. M., Turgut, A. E., & Tunc, L. T. (2022). A novel vision-based calibration framework for industrial robotic manipulators. *Robotics and Computer-Integrated Manufacturing*, *73*, 102248. https://doi.org/https://doi.org/10.1016/j.rcim.2021.102248

Bekkers, S. (2024). *Sensor fusion for visual-inertial simultaneous localisation and mapping* [Master's thesis, TU Delft]. http://resolver.tudelft.nl/uuid:06bf2239-db11-41ee-bab2-a90e95b3ceb8

Bekkers, S., Bettendorf, M. T., Reill, J., Giubilato, R., & Wedler, A. (2023). The lunar rover mini: Towards a versatile, open-source mobile robotic platform for educational and experimental purposes. *17th Symposium on Advanced Space Technologies in Robotics and Automation*. https://elib.dlr.de/202313/

Brockett, R. W. (1984). Robotic manipulators and the product of exponentials formula. In P. A. Fuhrmann (Ed.), *Mathematical theory of networks and systems* (pp. 120–129). Springer Berlin Heidelberg.

Caenen, J., & Angue, J. (1990). Identification of geometric and nongeometric parameters of robots. *Proceedings., IEEE International Conference on Robotics and Automation*.

Chin, K. W., von Konsky, B., & Marriott, A. (1997). Closed-form and generalized inverse kinematics solutions for the analysis of human motion. *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 'Magnificent Milestones and Emerging Opportunities in Medical Engineering' (Cat. No.97CH36136)*, *5*, 1911–1914 vol.5. https://doi.org/10.1109/IEMBS.1997.758709

Choi, Y., Cheong, J., Kyung, J. H., & Do, H. M. (2016). Zero-offset calibration using a screw theory. *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 526–528. https://doi.org/10.1109/URAI.2016.7625770

Daney, D., Papegay, Y., & Madeline, B. (2005). Choosing measurement poses for robot calibration with the local convergence method and tabu search. *The International Journal of Robotics Research*, *24*(6), 501–518. https://doi.org/10.1177/0278364905053185

De Luca, A., & Book, W. (2008). Robots with flexible elements. In *Springer handbook of robotics* (pp. 287–319). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_14

DLR. (2023). Surface avatar – an astronaut on board the iss controls a robot team on earth. https://www.dlr.de/en/latest/news/2023/03/surface-avatar-an-astronaut-on-board-the-iss-controls-a-robot-team-on-earth

DLR. (2024). Rafcon. https://dlr-rm.github.io/RAFCON/

Elatta, A., Gen, . L. P., Zhi, . F. L., Daoyuan, . Y., & Fei, . L. (2003). An overview of robot calibration. *Information Technology Journal*, *3*(1), 74–78. https://doi.org/10.3923/itj.2004.74.78

Gharaaty, S., Shu, T., Joubair, A., Xie, W. F., & Bonev, I. A. (2018). Online pose correction of an industrial robot using an optical coordinate measure machine system. *International Journal of Advanced Robotic Systems*, *15*(4), 172988141878791. https://doi.org/10.1177/1729881418787915

Hayati, S. (1983). Robot arm geometric link parameter estimation. *The 22nd IEEE Conference on Decision and Control*, 1477–1483. https://api.semanticscholar.org/CorpusID:45953383

He, R., Zhao, Y., Yang, S., & Yang, S. (2010). Kinematic-parameter identification for serial-robot calibration based on poe formula. *IEEE Transactions on Robotics*, *26*(3), 411–423. https://doi.org/10.1109/TRO.2010.2047529

Joubair, A., & Bonev, I. A. (2013). Comparison of the efficiency of five observability indices for robot calibration. *Mechanism and Machine Theory*, *70*, 254–265. https://doi.org/10.1016/j.mechmachtheory.2013.07.015

Joubair, A., & Bonev, I. A. (2014). Kinematic calibration of a six-axis serial robot using distance and sphere constraints. *The International Journal of Advanced Manufacturing Technology*, *77*(1–4), 515–523. https://doi.org/10.1007/s00170-014-6448-5

Kamali, K., Joubair, A., Bonev, I. A., & Bigras, P. (2016). Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra.2016.7487630

Kana, S., Gurnani, J., Ramanathan, V., Turlapati, S. H., Ariffin, M. Z., & Campolo, D. (2022). Fast kinematic re-calibration for industrial robot arms. *Sensors*, *22*(6), 2295. https://doi.org/10.3390/s22062295

Karan, B., & Vukobratović, M. (1994). Calibration and accuracy of manipulation robot models—an overview. *Mechanism and Machine Theory*, *29*(3), 479–500. https://doi.org/https://doi.org/10.1016/0094-114X(94)90130-9

Khalil, W., & Besnard, S. (2002). *Journal of Intelligent and Robotic Systems*, *34*(4), 357–379. https://doi.org/10.1023/a:1019687400225

Klodmann, J., Konietschke, R., Albu-Schaffer, A., & Hirzinger, G. (2011). Static calibration of the dlr medical robot miro, a flexible lightweight robot with integrated torque sensors. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/iros.2011.6095097

Kolyubin, S., Shiriaev, A., & Jubien, A. (2018). Consistent kinematics and dynamics calibration of lightweight redundant industrial manipulators. *The International Journal of Advanced Manufacturing Technology*, *101*(1–4), 243–259. https://doi.org/10.1007/s00170-018-2868-y

Kozlov, P., & Klimchik, A. (2022). Experimental study on robot calibration approaches. *IFAC-PapersOnLine*, *55*(10), 2785–2790. https://doi.org/10.1016/j.ifacol.2022.10.152

Labbé, M., & Michaud, F. (2018). Rtab map as an open source lidar and visual simultaneous localization and mapping library for large scale and long term online operation. *Journal of Field Robotics*, *36*(2), 416–446. https://doi.org/10.1002/rob.21831

Lattanzi, L., Cristalli, C., Massa, D., Boria, S., Lépine, P., & Pellicciari, M. (2020). Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task. *The International Journal of Advanced Manufacturing Technology*, *111*(7–8), 1813–1829. https://doi.org/10.1007/s00170-020-06179-9

Li, C., Wu, Y., Löwe, H., & Li, Z. (2016). Poe-based robot kinematic calibration using axis configuration space and the adjoint error model. *IEEE Transactions on Robotics*, *32*(5), 1264–1279. https://doi.org/10.1109/TRO.2016.2593042

Li, G., Zhang, F., Fu, Y., & Wang, S. (2019). Kinematic calibration of serial robot using dual quaternions. *Industrial Robot: the international journal of robotics research and application*, *46*(2), 247–258. https://doi.org/10.1108/ir-10-2018-0221

Liu, H., Zhu, W., Dong, H., & Ke, Y. (2018). An improved kinematic model for serial robot calibration based on local poe formula using position measurement. *Industrial Robot: An International Journal*, *45*(5), 573–584. https://doi.org/10.1108/ir-07-2018-0141

Matijevic, J. (1997). Sojourner: The mars pathfinder microrover flight experiment. https://dataverse.jpl.nasa.gov/dataset.xhtml?persistentId=hdl:2014/21704

METRIS. (2005). Metris k600 datasheet. https://www.metris3d.hu/k610_cmm_eu_0905.pdf

Mitchell, T. J. (1974). An algorithm for the construction of “d-optimal” experimental designs. *Technometrics*, *16*(2), 203. https://doi.org/10.2307/1267940

Motta, J. M. S., Llanos-Quintero, C. H., & Coral Sampaio, R. (2016). Inverse kinematics and model calibration optimization of a five-d.o.f. robot for repairing the surface profiles of hydraulic turbine blades. *International Journal of Advanced Robotic Systems*, *13*(3), 114. https://doi.org/10.5772/63673

NASA. (1988). Nasa facts: Viking mission to mars. https://mars.nasa.gov/internal_resources/828/

NASA. (1997). Nasa facts: Mars pathfinder. https://mars.nasa.gov/internal_resources/815/

NASA. (2024). Mars curiosity rover: Robotic arm. https://mars.nasa.gov/msl/spacecraft/rover/arm/

Flashforger. (2024). Acrylonitrile styrene acrylate. https://after-support.flashforge.jp/uploads/datasheet/tds/ASA_TDS_EN.pdf

Renders, J.-M., Rossignol, E., Becquet, M., & Hanus, R. (1991). Kinematic calibration and geometrical parameter identification for robots. *IEEE Transactions on Robotics and Automation*, *7*(6), 721–732. https://doi.org/10.1109/70.105381

Roth, Z., Mooring, B., & Ravani, B. (1987). An overview of robot calibration. *IEEE Journal on Robotics and Automation*, *3*(5), 377–385. https://doi.org/10.1109/jra.1987.1087124

Schmidt, F. (2023). Links and nodes. https://gitlab.com/links_and_nodes/links_and_nodes

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics*. Springer London. https://doi.org/10.1007/978-1-84628-642-1

Slamani, M., Nubiola, A., & Bonev, I. A. (2012). Modeling and assessment of the backlash error of an industrial robot. *Robotica*, *30*(7), 1167–1175. https://doi.org/10.1017/S0263574711001287

Subedi, D., Aune, T. N., Tyapin, I., & Hovland, G. (2022). Static deflection compensation of multi-link flexible manipulators under gravity. *IEEE Access*, *10*, 9658–9667. https://doi.org/10.1109/ACCESS.2022.3144404

Sugihara, T. (2011). Solvability-unconcerned inverse kinematics by the levenberg–marquardt method. *IEEE Transactions on Robotics*, *27*(5), 984–991. https://doi.org/10.1109/TRO.2011.2148230

Sun, Y., & Hollerbach, J. M. (2008). Observability index selection for robot calibration. *2008 IEEE International Conference on Robotics and Automation*, 831–836. https://api.semanticscholar.org/CorpusID:12867028

Tenhumberg, J., & Bauml, B. (2021). Calibration of an elastic humanoid upper body and efficient compensation for motion planning. *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. https://doi.org/10.1109/humanoids47582.2021.9555793

Wan, A., Song, L., Xu, J., Liu, S., & Chen, K. (2018). Calibration and compensation of machine tool volumetric error using a laser tracker. *International Journal of Machine Tools and Manufacture*, *124*, 126–133. https://doi.org/https://doi.org/10.1016/j.ijmachtools.2017.10.004

Wang, W., Wang, G., & Yun, C. (2014). A calibration method of kinematic parameters for serial industrial robots. *Industrial Robot: An International Journal*, *41*(2), 157–165. https://doi.org/10.1108/ir-06-2013-369

Wedler, A., Rebele, B., Reill, J., Suppa, M., Hirschmüller, H., Brand, C., Schuster, M., Vodermayer, B., Gmeiner, H., Maier, A., Willberg, B., Bussmann, K., Wappler, F., & Hellerer, M. (2015). Lru – lightweight rover unit. https://elib.dlr.de/102155/

Wu, L., Yang, X., Chen, K., & Ren, H. (2015). A minimal poe-based model for robotic kinematic calibration with only position measurements. *IEEE Transactions on Automation Science and Engineering*, *12*(2), 758–763. https://doi.org/10.1109/TASE.2014.2328652

Ziegert, J. C., & Datseris, P. (1988). Basic considerations for robot calibration. *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 932–938 vol.2. https://api.semanticscholar.org/CorpusID:19163100

A

Identification Jacobian Matrix Singular Values

This appendix displays the singular values obtained through the singular values decomposition (SVD) of the Identification Jacobian Matrix, which is composed of the derivatives of the position of the end effector with respect to the calibration parameters.

Singular Values				
2.34e+6	1.02e+3	8.45e+1	9.84e-1	5.31e-5
1.13e+6	8.16e+2	7.84e+1	7.31e-1	3.94e-5
8.07e+5	7.28e+2	6.74e+1	5.81e-1	3.45e-5
2.93e+5	6.16e+2	8.68	1.18e-4	3.22e-5
2.11e+5	4.79e+2	4.59	1.06e-4	1.35e-5
1.95e+4	3.37e+2	4.47	1.03e-4	8.79e-6
9.25e+3	2.43e+2	2.76	8.21e-5	9.92e-6
1.71e+3	1.41e+2	2.54	8.07e-5	
1.39e+3	1.09e+2	1.26	6.75e-5	

Table A.1: Table containing the singular values derived from the Singular Values Decomposition (SVD) of the Identification Jacobian Matrix built on the 43 calibration parameters.

B

Kinematic and Elastic Correction Parameters

This appendix shows the parameters obtained through the identification phase of the elasto-kinematic calibration for both sets of measurements: the ones with no load at the end effector and those with a load of 66.3 grams.

Param.	**Opt. 1**	**Opt. 2**	**Opt. 3**	**Opt. 4**	**Opt. 5**	**Opt. 1_2**	**Opt. 3_2**
$\delta\alpha_1$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta\alpha_2$ [rad]	-6.12e-3	-7.26e-3	0.00	-2.66e-3	0.00	-4.69e-3	0.00
$\delta\alpha_3$ [rad]	1.32e-2	1.77e-2	0.00	1.59e-2	0.00	1.65e-2	0.00
$\delta\alpha_4$ [rad]	8.71e-3	1.22e-2	0.00	1.90e-3	0.00	-2.03e-3	0.00
$\delta\alpha_5$ [rad]	-2.82e-2	-1.23e-1	0.00	-3.79e-5	0.00	1.74e-2	0.00
$\delta\alpha_6$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
k_0 [rad/Nm]	1.93e-2	0.00	2.23e-2	0.00	0.00	1.81e-2	2.40e-2
k_1 [rad/Nm]	7.57e-2	0.00	7.76e-2	7.88e-2	9.32e-2	7.71e-2	7.75e-2
k_2 [rad/Nm]	6.70e-2	0.00	8.24e-2	8.66e-2	9.55e-2	7.25e-2	8.04e-2
k_3 [rad/Nm]	7.68e-2	0.00	9.32e-2	8.94e-2	1.14e-1	8.22e-2	9.05e-2

k_4 [rad/Nm]	1.24e-1	0.00	1.48e-1	1.27e-1	1.41e-1	1.21e-1	1.40e-1
k_5 [rad/Nm]	1.04e-1	0.00	-4.84e-3	3.69e-1	5.42e-1	2.17e-1	9.51e-2
k_6 [rad/Nm]	-7.40e-3	0.00	6.27e-1	1.96e-1	6.23e-1	8.06e-3	5.08e-1
δa_1 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δa_2 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δa_3 [mm]	1.27e-1	-6.87e-1	0.00	-2.82e-2	0.00	4.87e-1	0.00
δa_4 [mm]	-5.34	3.31	0.00	-2.19	0.00	-3.85	0.00
δa_5 [mm]	4.09e-2	-6.71e-1	0.00	-1.22e-1	0.00	2.60e-2	0.00
δa_6 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta \theta_1$ [rad]	-3.45e-3	-1.87e-2	0.00	-5.70e-3	0.00	-4.82e-3	0.00
$\delta \theta_2$ [rad]	-2.32e-2	-1.30e-2	0.00	-2.30e-2	0.00	-2.44e-2	0.00
$\delta \theta_3$ [rad]	-2.96e-3	5.30e-4	0.00	5.42e-4	0.00	-1.86e-3	0.00
$\delta \theta_4$ [rad]	-1.11e-2	3.62e-2	0.00	3.65e-3	0.00	-3.81e-3	0.00
$\delta \theta_5$ [rad]	-2.32e-2	-4.92e-2	0.00	-1.94e-2	0.00	-2.56e-2	0.00
$\delta \theta_6$ [rad]	-1.83e-2	-4.27e-2	0.00	-2.43e-2	0.00	-2.65e-2	0.00
δd_1 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δd_2 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δd_3 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δd_4 [mm]	-1.66	-2.45	0.00	2.73e-1	0.00	5.84e-1	0.00

δd_5 [mm]	-1.90	7.48e-1	0.00	-1.42	0.00	-1.79	0.00
δd_6 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δx_b [mm]	-1.24	3.96	0.00	1.52	0.00	-2.07e-1	0.00
δy_b [mm]	-1.30	4.70e-1	0.00	-1.23	0.00	-1.40	0.00
δz_b [mm]	-5.31	-2.59e+1	0.00	-6.44	0.00	-4.78	0.00
$\delta \alpha_b$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta \beta_b$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta \gamma_b$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δx_t [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δy_t [mm]	-2.64	-7.86	0.00	-5.54e-3	0.00	5.28e-1	0.00
δz_t [mm]	1.55	-6.47e-1	0.00	1.39	0.00	1.51	0.00
$\delta \alpha_t$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta \beta_t$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\delta \gamma_t$ [rad]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δdd_3 [mm]	4.70	0.00	0.00	2.64	0.00	3.73	0.00
δdd_5 [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table B.1: Complete list of parameters derived from the parameter identification phase using the data collected from the measurement of 393 poses applying no load at the TCP. ”Opt.” stands for Option.

Parameters	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Option 5**
$\delta\alpha_1$ [rad]	0.00	0.00	0.00	0.00	0.00
$\delta\alpha_2$[rad]	-6.30e-3	-6.64e-3	0.00	-3.10e-3	0.00
$\delta\alpha_3$ [rad]	1.33e-2	1.74e-2	0.00	1.52e-2	0.00
$\delta\alpha_4$ [rad]	7.60e-3	1.00e-2	0.00	3.52e-4	0.00
$\delta\alpha_5$ [rad]	-4.07e-2	-1.25e-1	0.00	-2.39e-2	0.00
$\delta\alpha_6$ [rad]	0.00	0.00	0.00	0.00	0.00
k_0 [rad/Nm]	2.09e-2	0.00	2.29e-2	0.00	0.00
k_1 [rad/Nm]	7.11e-2	0.00	7.36e-2	7.34e-2	8.85e-2
k_2 [rad/Nm]	5.96e-2	0.00	7.45e-2	7.73e-2	8.70e-2
k_3 [rad/Nm]	6.56e-2	0.00	8.16e-2	7.88e-2	9.83e-2
k_4 [rad/Nm]	1.14e-1	0.00	1.35e-1	1.19e-1	1.32e-1
k_5 [rad/Nm]	1.29e-1	0.00	3.06e-2	3.83e-1	5.58e-1
k_6 [rad/Nm]	1.20e-1	0.00	4.39e-1	1.92e-1	5.09e-1
δa_1 [mm]	0.00	0.00	0.00	0.00	0.00
δa_2 [mm]	0.00	0.00	0.00	0.00	0.00
δa_3 [mm]	-8.26e-2	-1.08	0.00	-8.53e-2	0.00
δa_4 [mm]	-5.75	4.93	0.00	-1.63	0.00
δa_5 [mm]	1.15e-1	-8.21e-1	0.00	-2.80e-1	0.00

δa_6 [mm]	0.00	0.00	0.00	0.00	0.00
$\delta \theta_1$ [rad]	-2.78e-3	-2.02e-2	0.00	-5.43e-3	0.00
$\delta \theta_2$ [rad]	-2.34e-2	-1.43e-2	0.00	-2.29e-2	0.00
$\delta \theta_3$ [rad]	-5.71e-3	-5.11e-4	0.00	-7.42e-4	0.00
$\delta \theta_4$ [rad]	-1.45e-2	4.35e-2	0.00	5.50e-3	0.00
$\delta \theta_5$ [rad]	-1.52e-2	-5.27e-2	0.00	-1.04e-2	0.00
$\delta \theta_6$ [rad]	-2.05e-2	-4.57e-2	0.00	-2.49e-2	0.00
δd_1 [mm]	0.00	0.00	0.00	0.00	0.00
δd_2 [mm]	0.00	0.00	0.00	0.00	0.00
δd_3 [mm]	0.00	0.00	0.00	0.00	0.00
δd_4 [mm]	-1.49	-2.06	0.00	5.75e-1	0.00
δd_5 [mm]	-1.77	7.24e-1	0.00	-1.45	0.00
δd_6 [mm]	0.00	0.00	0.00	0.00	0.00
δx_b [mm]	-1.70	4.20	0.00	1.46	0.00
δy_b [mm]	-1.11	8.86e-1	0.00	-1.00	0.00
δz_b [mm]	-5.43	-2.94e+1	0.00	-7.43	0.00
$\delta \alpha_b$ [rad]	0.00	0.00	0.00	0.00	0.00
$\delta \beta_b$ [rad]	0.00	0.00	0.00	0.00	0.00
$\delta \gamma_b$ [rad]	0.00	0.00	0.00	0.00	0.00

δx_t [mm]	0.00	0.00	0.00	0.00	0.00
δy_t [mm]	-3.54	-7.87	0.00	-1.24	0.00
δz_t [mm]	1.60	-6.58e-1	0.00	1.42	0.00
$\delta \alpha_t$ [rad]	0.00	0.00	0.00	0.00	0.00
$\delta \beta_t$ [rad]	0.00	0.00	0.00	0.00	0.00
$\delta \gamma_t$ [rad]	0.00	0.00	0.00	0.00	0.00
δdd_3 [mm]	4.81	0.00	0.00	2.92	0.00
δdd_5 [mm]	0.00	0.00	0.00	0.00	0.00

Table B.2: Complete list of parameters derived from the parameter identification phase using the data collected from the measurement of 393 poses applying a load of 66.3 g at the TCP.