BSc Thesis

A Robotic Approach to Antenna Measurement: Inverse Kinematics, Path Planning and Placement Optimization

H.A.W. Maas
F.T. Wichman

Delft University of Technology

TU Delft Delft University of Technology

THz Sensing Group

Challenge the future

This page intentionally left blank.

A Robotic Approach to Antenna Measurement: Inverse Kinematics, Path Planning and Placement Optimization

by

H.A.W. Maas
F.T. Wichman

to obtain the degree of Bachelor of Science
at the Delft University of Technology,

Student number:	5858038 (Huib Maas)	
Student number:	5792428 (Fabian Wichman)	
Project duration:	April, 2025 – June, 2025	
Supervisors:	dr. D. Cavallo,	TU Delft
	Ir. J. Geng,	TU Delft
	Prof. dr. N. Llombart,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

This thesis explores the automation of Near-Field Region (NF) and Far-Field Region (FF) antenna measurements using a 6 Degrees of Freedom (DoF) robotic arm. A path planning algorithm is developed to detect and minimize motion discontinuities ("jumps") by evaluating joint movements through a cost function. An Inverse Kinematics (IK) method is used to find all possible joint configurations, allowing for smoother path plannings for the measurements. Various sorting algorithms are developed and tested on both planar and spherical grids to determine the most efficient measurement paths in terms of path smoothness. Furthermore, the optimum placement of the Antenna Under Test (AUT) is studied using the above-mentioned method, to ensure path feasibility while minimizing the strain on the probe cable. Experimental validation is performed in MATLAB, commercial simulation software RoboDK, and on the physical robotic arm. The results confirm that the proposed algorithms successfully reduce joint jumps and cable strain, enabling accurate and automated antenna measurements in both NF and FF configurations.

Preface

Over the past two months, we have investigated the possibilities of using a robotic arm for automated antenna measurements as part of the Bachelor Graduation Project (BEP). Together with Guus Hak, Tom Rietjens, Sven Lemsom, Sitti Romijn, Danylo Zavoloko, and Matthieu Gillain, we automated the measurement systems for both near-field (NF) and far-field (FF) tests using a robotic arm and several optimization techniques. We would like to thank them for their efforts and the enjoyable collaboration.

We are especially grateful to our daily supervisor, ir. Jingling Geng, for always being ready to help, and to our supervisors, prof. dr. N. Llombart and dr. D. Cavallo, for their valuable feedback and advice.

Finally, we thank our families for supporting us throughout the past three years of our Bachelor's in Electrical Engineering, granting us the opportunity and time to pursue this path. Without their support, this graduation project would not have been possible.

H.A.W. Maas
F.T. Wichman
Delft, June 2025

Contents

Glossary

AUT Antenna Under Test. i, v, 2, 3, 8, 18–23, 25

Cobots Collaborative Robots. 1

DHm Modified Denavit Hartenberg. 7

DoF Degrees of Freedom. i, 1, 6, 25

FF Far-Field Region. i, 1, 2, 8, 9, 25, 26

IK Inverse Kinematics. i, 6, 7, 25

NF Near-Field Region. i, 1, 2, 4, 8, 9, 25, 26

List of Figures

1 Introduction

Antennas are crucial components in modern communication and sensing systems which are being explored at higher frequencies (mm-wave) for larger bandwidths and compactness. The properties of the antennas plays a critical role in determining the overall system performance. Therefore, it is of great importance to accurately characterize the performance of these antennas, in terms of parameters such as radiation pattern, directivity, gain, impedance and bandwidth [1].

Conventional antenna measurements are typically performed in controlled environments like anechoic chambers using static or manually adjusted configurations [2], [3]. Although these approaches provide reliable results, they are often limited in terms flexibility and alignment accuracy for evaluating modern high-frequency antennas. Moreover, there is a trend of using higher frequencies [4], [5]. Using a higher frequency for 6G, for example, a larger bandwidth can be obtained for wireless communication systems. The downside of the higher frequency is the requirement for the accuracy of measurements. The alignment requirement for the probe is in the range of about $\lambda/100$ to $\lambda/50$ for planar scanning [2]. For a high frequency antenna at $100 - 300$GHz this results in a accuracy of $60 - 20\mu$m [6].

With the growing adoption of Collaborative Robots (Cobots) across various sectors, including industry and healthcare, these systems are increasingly used to perform tasks more quickly, precisely, safely and with reduced risk of failure [7]–[9]. In antenna measurement setups, six-axis robotic arms are favored for their high precision and mobility, offering six DoF that enable flexible scanning patterns, such as planar or spherical, with improved repeatability and speed [10]–[13].

With excessive DoF of the robotic arm, careful joint configuration setting and path planning are essential to ensure smooth motion during measurements. This prevents cable twisting and mitigates the risk of damaging sensitive measurement equipment.
This thesis presents the development of the possibilities of measuring an antenna using a six-axis robotic arm in NF and FF, where the robotic arm operates in planar or spherical geometries without significant jumps in the movement of the robotic arm. Experimental results are obtained in theoretical (software) and practical experiments in which the FANUC CRX-20iA/L is used as the robotic arm. This includes optimizing the measurement path, determining the optimal placement of the antenna, and integrating sorting algorithms to ensure efficient movement.

1.1. Project Overview

The complete measurement system must be able to perform measurements on an AUT in NF and FF. These measurements can give insights in the characteristics of the AUT. To automate antenna radius measurements using a robotic arm, the system is divided into four subsystems, one for every subgroup (Table 1.1).

Table 1.1: Subgroup responsibilities.

Subgroup 1	Subgroup 2	Subgroup 3	Subgroup 4
Programming and automation of robotic arm	Control interface for robotic arm and network analyzer	Far-field antenna pattern measurements (spherical acquisition)	Near-field antenna pattern measurements (planar acquisition)

Subgroup 1 is responsible for programming the robotic arm (see Figure 1.1) to ensure a smooth path for measurements is taken together with optimal antenna placement and sorting algorithms. Subgroup 1 uses the grid generated by subgroup 3 (spherical) and subgroup 4 (planar).

Figure 1.1: Illustrative image for subgroup 1.

Subgroup 2 is responsible for the GUI, an interface where the user can select the input parameters and can see the measurement data. This GUI connects the code of the four subgroups. Subgroup 2 also sends commands to the VNA (Vector Network Analyzer) to take measurements when a point on the grid is reached (Figure 1.2).

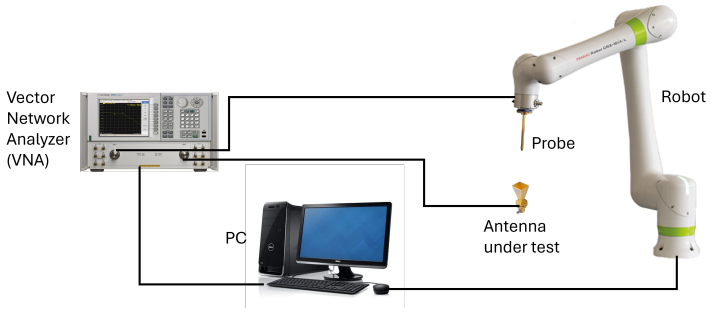


Figure 1.2: Illustrative image for subgroup 2.

Subgroup 3 is responsible for generating the grid for spherical field (Figure 1.3) and to post-process the obtained data.

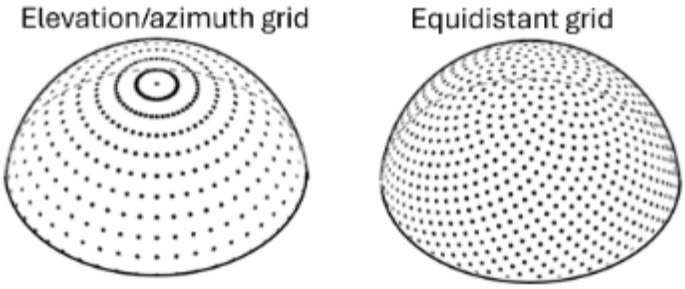


Figure 1.3: Illustrative image for subgroup 3.

Subgroup 4 is responsible for generating the grid for planar field (Figure 1.4) and to post-process the obtained data.

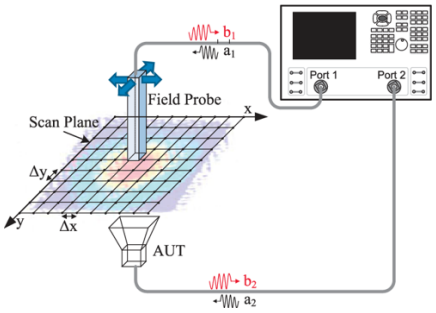


Figure 1.4: Illustrative image for subgroup 4.

1.2. Thesis Synopsis

Chapter 2 presents the theoretical background relevant to the project. In chapter 3, the requirements that serve as a reference for defining the objectives of this work are introduced. In chapters 4 and 5 the method is given of automating the robotic design, where discontinuities (or "jumps") in the robotic arm's joint movements are analyzed. These discontinuities are then used to assign a cost to each measurement path, serving as a metric for evaluating path smoothness. Various sorting algorithms are tested to assess their impact on this path cost. In chapter 5, the influence of the antenna placement (AUT) on the system performance is examined and results are presented in chapter 6. Finally, chapter 7 presents an analysis of the results and reflects on the requirements.

2

Theoretical Background

2.1. Polarization

Antennas are typically designed to operate with a specific type of polarization. The antenna polarization refers to the orientation of the radiated electric field. To ensure optimal performance, maintaining a high degree of polarization purity within the main beam is crucial. This is usually achievable in antennas with very high directivity. On the other hand, antennas with lower directivity tend to exhibit higher levels of cross-polarization across a wider range of angles [14]. Cross-polarization is the polarization orthogonal to a specified reference polarization, usually called the co-polarization [15]. This unwanted cross-polarized radiation reduces the strength of the desired signal because it diverts part of the input energy away from the main polarization. Therefore, it is important to measure both polarizations of the electric field radiated by an antenna.

There are multiple ways to describe these polarizations, depending on the source and the measurement setup (planar or spherical scan). Ludwig (1973) [16] defines three different cases for the co- and cross-polarization (see Figure 2.1). The unit vectors ($\hat{u}_{co}$ and $\hat{u}_{cross}$, respectively) are shown in the figure.

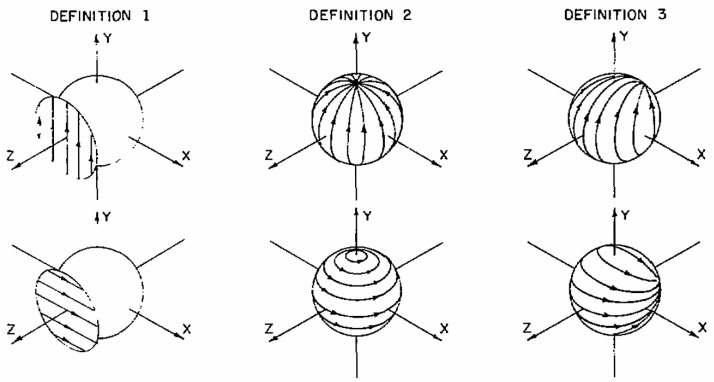


Figure 2.1: Polarization definitions of Ludwig. Top: direction of co-polarization. Bottom: direction of cross-polarization. [16]

The three cases are defined as follows [14]:

- **Ludwig 1** (Planar): This definition describes the unit vectors of a rectangular plane (see Figure 2.1), which is usually used for a NF planar scan. The co- and cross-polarization depend on the orientation of the antenna. The polarization of the electric field of the antenna gives the direction of the co-polarization:

  - Y polarized antenna, that is, the E-field of the antenna is in the y direction:

$$\hat{u}_{co_{L1-y}} = \hat{y} \tag{2.1}$$

$$\hat{u}_{cross_{L1-y}} = \hat{x} \tag{2.2}$$

  - X polarized antenna:

$$\hat{u}_{co_{L1-x}} = \hat{x} \tag{2.3}$$

$$\hat{u}_{cross_{L1-x}} = \hat{y} \tag{2.4}$$

- **Ludwig 2** (Spherical): This definition describes a sphere with unit vectors tangential to the surface. The co- and cross-polarization unit vectors correspond to the θ and ϕ directions of a perfectly linear-polarized antenna, rotated such that θ=0 and ϕ=0 corresponds to the positive y- or x-axis, instead of the positive z-axis [14]. The second definition of co-polarization corresponds to the field radiated by a short dipole oriented along x or y. In other words, an ideal electric dipole would have zero cross-polarization in every direction according to this definition. Similar to the previous case:

  - Y-polarized antenna: the unit vectors have to be rotated so that the co-polarization lines come out of the y axis (see Figure 2.1). This gives the following definition, where:

$$\hat{u}_{co_{L2-y}} = \frac{\cos\phi\cos\theta\hat{\theta} - \sin\phi\hat{\phi}}{\sqrt{1-\sin^2\phi\sin^2\theta}} \tag{2.5}$$

$$\hat{u}_{cross_{L2-y}} = \frac{\sin\phi\hat{\theta} + \cos\phi\cos\theta\hat{\phi}}{\sqrt{1-\sin^2\phi\sin^2\theta}} \tag{2.6}$$

  - X-polarized antenna:

$$\hat{u}_{co_{L2-x}} = \frac{\sin\phi\cos\theta\hat{\theta} + \cos\phi\hat{\phi}}{\sqrt{1-\cos^2\phi\sin^2\theta}} \tag{2.7}$$

$$\hat{u}_{cross_{L2-x}} = \frac{\cos\phi\hat{\theta} - \sin\phi\cos\theta\hat{\phi}}{\sqrt{1-\cos^2\phi\sin^2\theta}} \tag{2.8}$$

- **Ludwig 3** (Spherical): This definition is widely regarded as the most suitable for describing antenna patterns, particularly in systems where the antenna can be rotated about the ϕ direction, as the unit vectors remain consistently aligned with the measurement configuration [16]. The unit vectors are obtained by rotating $\hat{\phi}$ and $\hat{\theta}$ about the radial direction by the angle ϕ (see Figure 2.1). The third Ludwig's definition of co-polarization corresponds to the field radiated by a Huygens source, which is the combination of a short electric dipole and an orthogonal magnetic dipole [14]. This gives the following definitions:

  - Y-polarized antenna:

$$\hat{u}_{co_{L3-y}} = \sin\phi\hat{\theta} + \cos\phi\hat{\phi} \tag{2.9}$$

$$\hat{u}_{cross_{L3-y}} = \cos\phi\hat{\theta} - \sin\phi\hat{\phi} \tag{2.10}$$

  - X-polarized antenna:

$$\hat{u}_{co_{L3-x}} = \cos\phi\hat{\theta} - \sin\phi\hat{\phi} \tag{2.11}$$

$$\hat{u}_{cross_{L3-x}} = \sin\phi\hat{\theta} + \cos\phi\hat{\phi} \tag{2.12}$$

Consequently, the robot arm's motion must adapt to the polarization type being measured. Since each polarization definition demands a specific probe orientation, the control algorithm must conform with these variations to ensure accurate measurements.

2.2. Inverse Kinematics (IK)

Since there are limits on the six axis of the robotic arm, the robotic arm cannot turn continuously, as shown in Table 2.1. This results in jumps, where the robotic arm has to completely change configuration in an unwanted and unsafe manner before reaching a point. This could interfere with the measurements or even damage the robotic arm or the test equipment itself.

Table 2.1: Motion range of the joints [°].

J_1	J_2	J_3	J_4	J_5	J_6
360	360	540	380	360	450

Using the solutions of the IK equations, a set of possible configurations of the joints to reach the desired pose, is obtained. For 6 DoF robots with a spherical wrist, an IK model can be defined, for which up to eight solutions exist [17]. Since a robotic arm of the CRX series (FANUC CRX-20iA/L) with a non-spherical wrist is used, this IK model cannot be applied. Figure 2.2 shows the used robotic arm and the connections of the six joints, labeled from J_1-J_6. This particular robotic arm has perpendicular axes for J_2 and J_3, which makes the analytical solution of the IK equations unavailable.

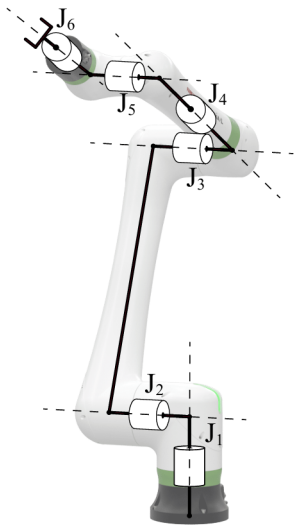


Figure 2.2: Kinematic architecture of collaborative robot FANUC CRX-20iA/L.

In 2024, Abbes uses a CRX-10iA/L to determine the IK solutions using an algebraic approach [18]. Note that this approach is also applicable to the CRX-20iA/L, since the 6 DoF, joints configuration and limits are the same. In this approach, the configuration of the joints parameters is defined as

$$[J] = [J_1, J_2, J_3, J_4, J_5, J_6]^T \tag{2.13}$$

where the position is denoted by

$$P = FK(J) = [X, Y, Z, W, P, R]^T \tag{2.14}$$

with (X, Y, Z) the coordinates of the end-effector center (or Tool Center Point), and (W, P, R) the three rotation angles with Cardan's approach to characterize the end-effector orientation. These three angles are, respectively, measured along the three axes x, y, and z of the robot reference frame R_0, which is at the origin of the robotic arm. Here FK(J) is the Forward Kinematics (FK) solution for the given joints, i.e. the pose reached when the joints are configured as J. The FANUC convention states that:

- W, for "yaW" angle, is measured around the X-axis;
- P, for "Pitch" angle, is measured around the Y-axis;
- R, for "Roll" angle, is measured around the Z-axis.

In this approach, the Modified Denavit Hartenberg (DHm) approach is used as explained by [19]–[21], which states that four parameters are used to characterize the relative position and orientation between consecutive links (joints). This simplifies the derivation of the IK equations. The four DHm parameters are:

- Rotation of an angle α_{i-1} around X_{i-1};
- Translation of α_{i-1} along X_{i-1};
- Rotation of an angle θ_i around Z_i;
- Translation of r_i along Z_i.

These parameters define the transformation from frame $i-1$ to frame i, and together they capture the relative orientation and position between two links. In this work, the DHm parameters are applied to the FANUC CRX-20iA robot using the joint conventions defined by FANUC. Notably, the angle θ_3 is expressed as $J_2 + J_3$ to reflect the mechanical coupling implemented in the robot's control system: moving joint 2 automatically affects joint 3 to maintain a consistent wrist orientation.
The table below summarizes the DHm parameters. Each row corresponds to one of the four DHm parameters, and each column to a link or joint in the robot's kinematic chain.

Table 2.2: DHm Parameters: lengths (mm), angles (deg). J_i, the joint values, are expressed following the same convention as FANUC [8].

Link	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**
a_{i-1}	0	0	540	0	0	0
α_{i-1}	0	-90	+180	-90	+90	-90
θ_i	J_1	$J_2 - 90$	$J_2 + J_3$	J_4	J_5	J_6
r_i	0	0	0	-540	150	-160

There are at most 16 possible configurations in which the cobot reaches the desired pose and orientation [18]. However, for 80% of the configurations, there are only 8 generic solutions due to double solutions and inaccuracies of the reached pose with respect to the desired pose [18]. In some cases, the number of solutions is limited due to the angular limits of the joints and their reach.

3

Program of Requirements

3.1. General Description of the System

The product to be developed is a measurement system for automating antenna measurements of the AUT in both NF and FF. The system includes a software framework and a robotic arm subsystem. It is intended for use in a professional lab environment where flexibility, precision, and automation are essential. This Program of Requirements (PoR) outlines the conditions and performance criteria under which the product must be developed, implemented, and evaluated. The requirements are categorized according to the Capability Maturity Model Integration (CMMI) and the MoSCoW method and classified as:

Mandatory Requirements (MR): The system must meet these for acceptance.
Trade-off Requirements (ToR): These are desirable features that improve the value of the system and may be balanced against each other.

All requirements are grouped into **functional** and **non-functional** categories.

3.2. Requirements for the Complete System

3.2.1. Functional Requirements

A-F.1 **Must:** Do antenna radius measurements of the AUT in both NF and FF.
A-F.2 **Must:** Support measurement of a wide range of antenna types and sizes.
A-F.3 **Must:** Allow users to choose or adjust input settings.
A-F.4 **Must:** Achieve measurement precision.
A-F.5 **Should:** Generate graphical figures summarizing measurement results.

3.2.2. Non-Functional Requirements

A-NF.1 **Must:** Implemented using MATLAB.
A-NF.2 **Should:** Provide a user-friendly interface with clear navigation and feedback.
A-NF.3 **Should:** Ensure the controller interface is intuitive and requires minimal user training.
A-NF.4 **Could:** Include comprehensive user documentation and help resources.

3.2.3. Trade-Off Requirements

- The execution speed of visualizations (ToR A-F.5) may trade off with graphical resolution.
- The intuitiveness of the interface (ToR A-NF.2 and A-NF.3) may be traded for the number of features.
- The scope of documentation (ToR A-NF.4) can trade off with development time and cost.

3.3. Requirements for the Robotic Arm Subsystem

This subsystem automates radial antenna measurements and must comply with safety, precision, and flexibility criteria.

3.3.1. Functional Requirements

B-F.1 **MR:** The robotic arm must use X,Y,Z (matrix $Nx3$) coordinates given by subgroup 4 for planar (NF) scans.

B-F.2 **MR:** The robotic arm must use r, θ, ϕ (matrix $Nx3$) coordinates given by subgroup 3 for spherical (FF) scans.

B-F.3 **MR:** The robotic arm must support spherical scans using both Ludwig II and Ludwig III polarization definitions.

B-F.4 **MR:** The robotic arm must be able to follow a smooth path.

B-F.5 **MR:** The robotic arm must perform antenna pattern measurements with an angular resolution of 1° or finer.

B-F.6 **MR:** The robotic arm must be able receive 'move-to' commands from the control system of subgroup 2 in the form of joint-configuration vectors.

B-F.7 **MR:** The robotic arm must send a digital readiness signal (logic HIGH) when it reaches a target position within 0.1 mm tolerance.

B-F.8 **ToR:** The robotic arm could allow adjustable motion speeds.

3.3.2. Non-Functional Requirements

B-NF.1 **MR:** The robotic arm control algorithms must be implemented in MATLAB.

B-NF.2 **ToR:** The software should be modular to allow future upgrades with minimal redesign.

3.3.3. Trade-Off Requirements

- Allowing a wide adjustable motion speed range (ToR B-F.8) may trade off with safety and motion precision (MRs B-F.5 and B-F.7), particularly at higher speeds.
- System modularity for future upgrades (ToR B-NF.2) may trade off with system optimization and integration needed for minimal motion latency (ToR B-NF.1).

This PoR provides a precise, measurable framework for evaluating design choices and implementation outcomes throughout the thesis. All mandatory requirements must be fulfilled. Trade-off requirements guide optimization and offer flexibility based on practical constraints.

4

Path Planning

4.1. Introduction

To ensure smooth and accurate motion, the robot arm must follow an optimized sequence through all measurement points. This sequence is determined using path planning algorithms that minimize jumps. Such jumps can lead to safety issues or affect measurement quality. A cost-based evaluation is used to quantify the impact of these jumps and compare different path options. Several sorting methods are applied to both planar and spherical grids to determine the most efficient path. From one point to the next, we calculate all the possible solutions and choose the one with minimum cost (minimum joint movement). Additionally, the probe's offset is accounted for to maintain positional accuracy after mounting the probe.

4.2. Joint Movement Discontinuity: Jumps

To determine whether the path followed by the robotic arm is smooth, the number of jumps can be obtained. A jump is a discontinuity in the movement of the robotic arm and is caused by the limits of the individual joints of the robotic arm. Since these joints cannot move for a full 360° continuously, the joint can reach a limit and must move the other way to change orientation. Moreover, when such a reconfiguration takes place, the joint makes a sudden jump in angle, which can damage the equipment or even harm the user. Especially $J_1 - J_3$ are dangerous because of their major movements. To minimize this behavior and smooth the path of the robotic arm, the number of jumps can be obtained by using the derivative of the joint movement.

4.2.1. Derivation of a Jump

For each target, there are multiple joint configurations to reach the desired pose (chapter 2). This results in several paths, i.e. sequences through the target points with unique movements of the joints. To determine the number of jumps in this solution, the derivative of the movement of each six joints is used. This derivative, with respect to the targets, indicates how much each joint has to rotate to reach the next point. A jump occurs when the joint has to rotate disproportionately. To identify whether a jump is disproportionately, an individual threshold is used (Equation 4.1).

$$\text{Threshold} = [80, 80, 80, 80, 80, 90] \tag{4.1}$$

When the peaks of the derivative exceed this threshold, jumps occur. For joint 6, a threshold of 90° is used, since this joint always has to move 90° to account for the change in polarization. By setting this threshold at 90°, this jump is not taken into account. The other thresholds are chosen, such that the robotic arm does not rotate too much and tested to ensure safety of the measurement equipment.

4.2.2. Results

In Figure 4.1 and Figure 4.2, the derivation of the number of jumps can be seen. Both solutions use a spherical grid based on Ludwig III with a radius of 150mm, centered at a elevation angle $\theta_0 = 60°$, and sampled at 100 points arranged in a 10×10 ($\theta \times \phi$) grid. In the case of Figure 4.1, there are jumps in $J_1 - J_3$, so the solution is declared invalid. For Figure 4.2, there are no jumps in $J_1 - J_3$, so the solution is valid.

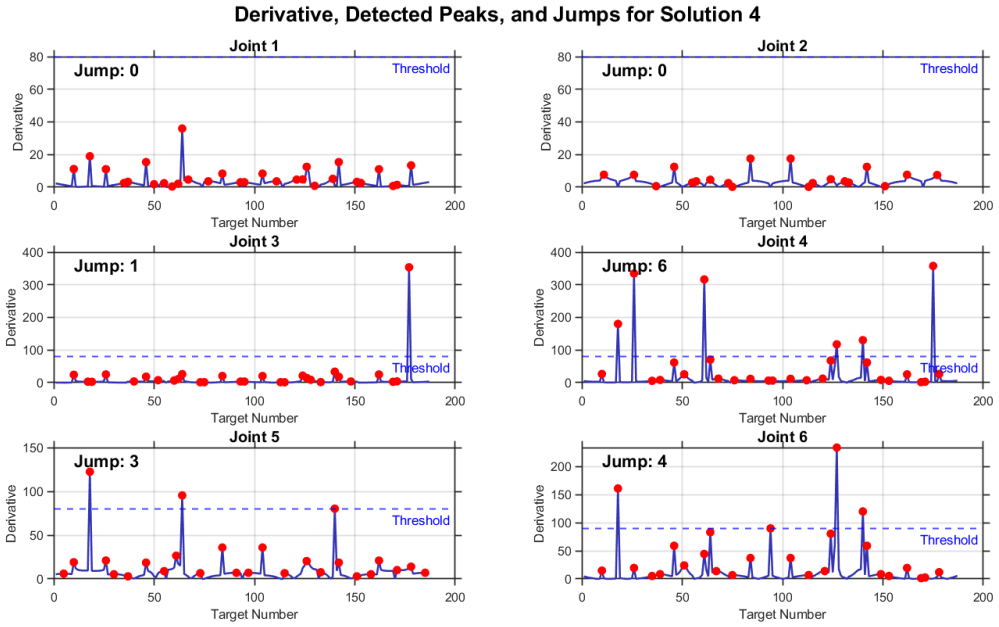


Figure 4.1: Derivation of jumps for an invalid solution.

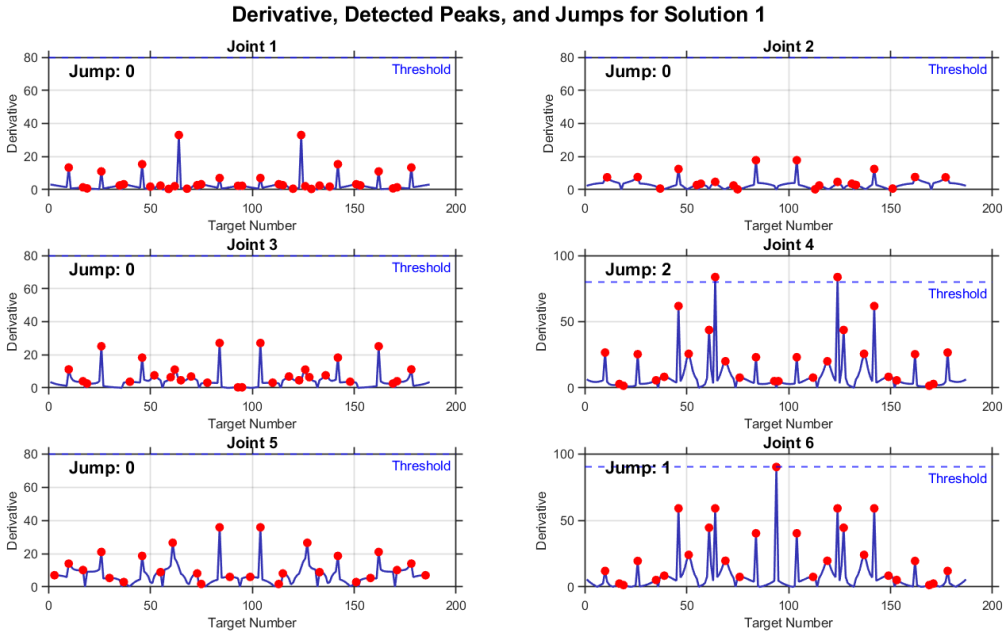


Figure 4.2: Derivation of jumps for a valid solution.

The code for this derivation can be seen in Appendix B.

4.3. Definition of Cost Function

To ensure a smooth path, it is desirable to minimize the number of jumps. Examining the kinematic architecture of the robot arm (Figure 2.2), it becomes clear that certain joints (J_1-J_3) control larger sections of the arm. When jumps occur in these joints, they can result in significant and abrupt movements, which may pose a risk to both the surroundings and the accuracy of the measurement. However, for joints that control a smaller part of the arm, a jump would be less dangerous (J_4-J_6).

To quantify the impact of jumps, a cost algorithm is implemented. Each joint is assigned a specific cost value, and whenever a jump occurs at that joint, i.e. the rotation of the joints exceeds the threshold (Equation 4.1), its corresponding cost is added to the total. The smoothest path is the one with the lowest cost. The six joints of the arm are chosen to be assigned with the following cost:

- J_1-J_3: 1000
- J_4: 50
- J_5: 5
- J_6: 1

If the total cost exceeds 1000, the path is considered not smooth and is deemed invalid. This threshold is set because jumps in joints J_1 to J_3, which control larger sections of the arm, could lead to potentially dangerous movements. While jumps in J_4 and J_5 are undesirable, they are not critical; however, if a path exists without any jumps at J_4 and J_5, it is preferred. Joint J_6 has minimal impact on the overall motion, as it simply rotates the probe and are therefore less significant in the evaluation.

The probe connected to the robot includes a cable that must also be mounted on the robotic arm. This is done using straps to secure the cable along the arm (Figure 4.3). However, this setup introduces certain constraints: excessive rotation of a joint can cause the cable to twist, potentially leading to damage and disrupting the measurement. Preventing this efficiently is challenging, as the exact behavior of the cable is difficult to predict. To mitigate the risk, the total angular displacement of J_6 is calculated in advance by subtracting the minimum displacement from the maximum displacement. If this displacement exceeds a certain threshold, the corresponding path is deemed invalid. Note that only J_6 is used, as this joint has the greatest impact on the cable.

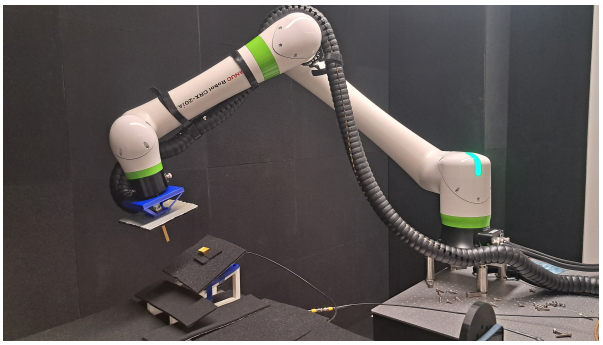


Figure 4.3: Cable safety limit in lab.

A simplified overview of the cost algorithm is given in the figure below (Figure 4.4). The code for this cost algorithm is given in Appendix B.

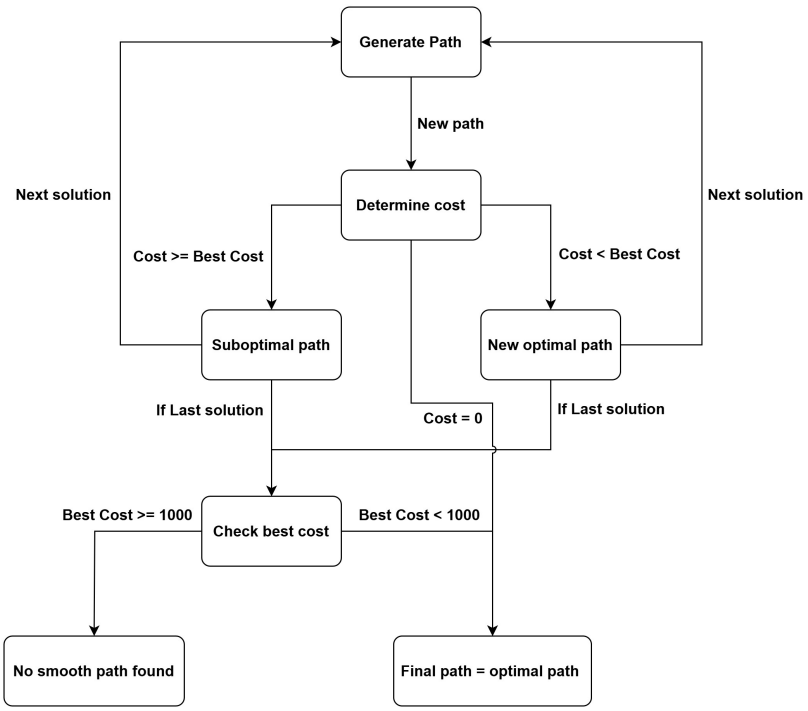


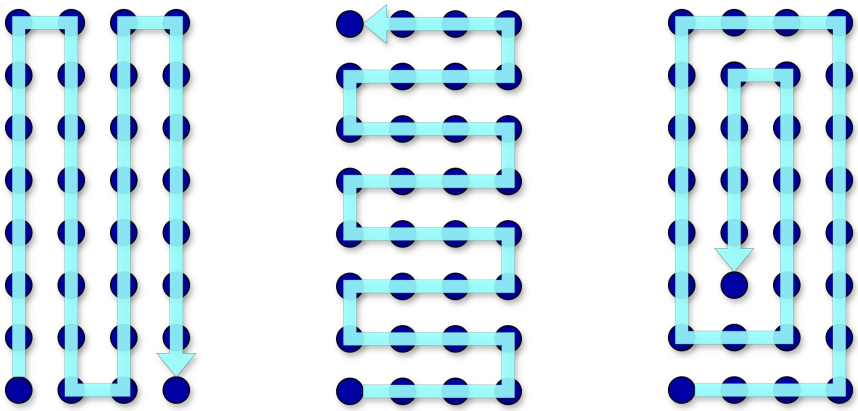
Figure 4.4: Cost algorithm.

4.4. Sorting Algorithm

To make the robot arm move as smooth as possible different algorithms were used to sort the given grid points. This determines the path the robot arm would take to go through all points. Multiple sorting algorithms for both planar and spherical grids were designed.

4.4.1. Planar Scan

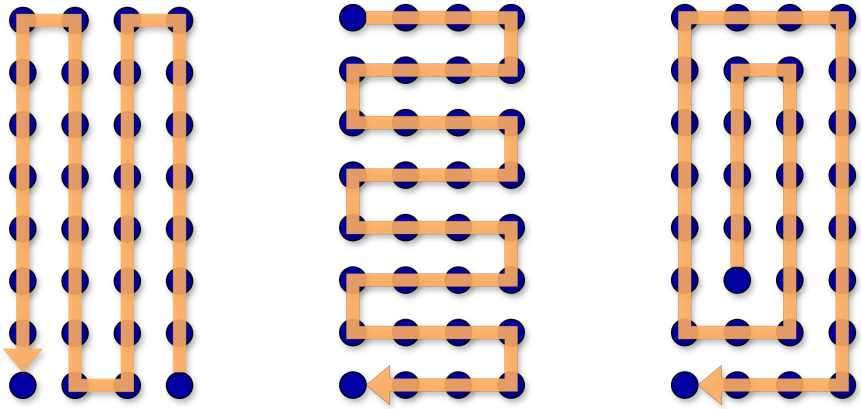
- **Snake-long:** The robot arm moves over the long axis, going back and forth while moving over the short axis after each "row" (see Figure 4.5a).
- **Snake-short:** The robot arm moves over the short axis, going back and forth while moving over the long axis after each "row" (see Figure 4.5b).
  These algorithms are only different if the grid is rectangular. If the Planar grid is a square, the two algorithms will provide the exact same sorting.
- **Spiral:** The robot arm moves in a spiral, moving counter-clockwise while converging to the middle, as shown in Figure 4.5.



(a) Snake long. (b) Snake short. (c) Spiral.

Figure 4.5: Co-polarization for planar.

- **Polarization:** For all three algorithms, after completing the path the probe will rotate 90° (switch polarization) and the robot arm will return along the same path in the reverse direction (see Figure 4.6).



(a) Snake long. (b) Snake short. (c) Spiral.

Figure 4.6: Cross-polarization for planar.

Tilted Planar Scan

For a tilted planar scan, the input from subgroup 4 consists of the rotated target points, along with the rotation parameters θ and ϕ. These parameters are used to rotate the coordinate reference system accordingly, aligning it with the robot's orientation. Importantly, the reference frame maintains the original polarization directions, ensuring consistent co- and cross-polarization. This is done by rotating the local $\hat{x}$ around the Y-axis by the angle ϕ:

$$\hat{x}' = \cos\theta\hat{x} - \sin\theta\hat{z} \tag{4.2}$$

Following, the local $\hat{z}$ is rotated about both θ and ϕ:

$$\hat{z}' = -\sin\theta\cos\phi\,\hat{x} - \sin\theta\sin\phi\,\hat{y} - \cos\theta\,\hat{z} \tag{4.3}$$

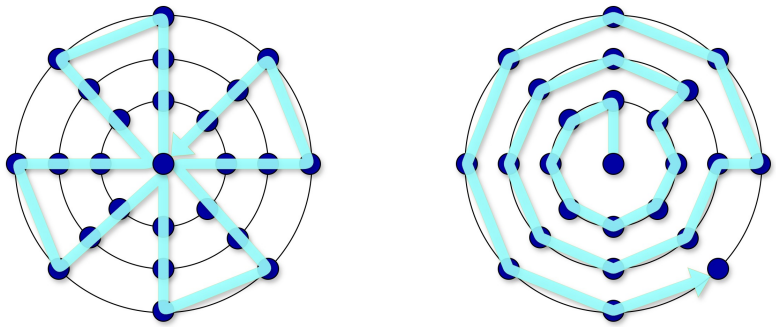
Then $\hat{y}$ is the result of the cross product of $\hat{x}$ and $\hat{z}$:

$$\hat{y}' = \hat{z}' \times \hat{x}' \tag{4.4}$$

The sorting algorithm remains unchanged, it determines the long and short axes of the grid and then sorts the points accordingly.

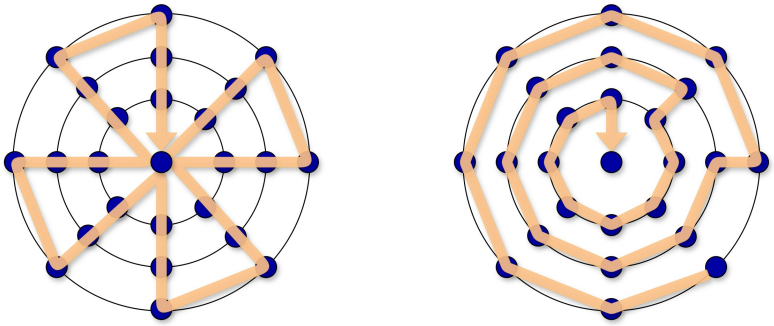
4.4.2. Spherical Scan

- **Snake:** The robot arm moves over each cut (multiple θ for one ϕ), going back and forth while increasing in ϕ after each cut (see Figure 4.7a).
- **Circular:** The robot arm moves over each circle (multiple ϕ for one θ), while increasing in θ after each circle (see Figure 4.7b).
  For these two algorithms the probe will, after completing the whole path, rotate 90° (switch polarization) and the robot arm will return along the same path in the reverse direction (see Figure 4.8).
- **Theta-cuts:** The robot arm moves along a single cut, switches polarization at the end, and returns along the same path. It then proceeds to the next cut and repeats the process (see Figure 4.9).
- **Phi-cuts:** The robot arm moves along a single circle, switches polarization at the end, and returns along the same path. It then proceeds to the next circle and repeats the process (see Figure 4.10).



(a) Snake. (b) Circular.

Figure 4.7: Co-polarization for spherical.



(a) Snake. (b) Circular.

Figure 4.8: Cross-polarization for spherical.

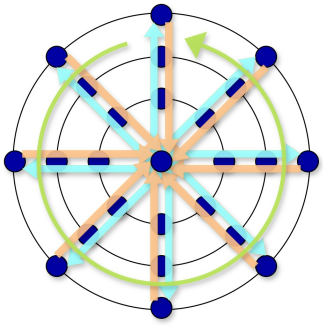


Figure 4.9: Theta cuts.

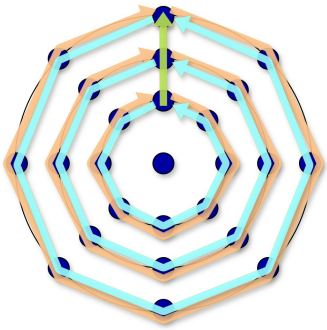


Figure 4.10: Phi cuts.

4.5. Probe Offset

It has to be taken into account that the probe is also attached to the robotic arm. The end of this probe is where the actual measurement is done. So there is a change in the local z, and even local x and y if the probe is not in the middle of the robotic arm. This "probe offset" is given by the user and will be taken into account when performing the calculations for the path. The offset can also be seen in simulation (see Figure 4.11).

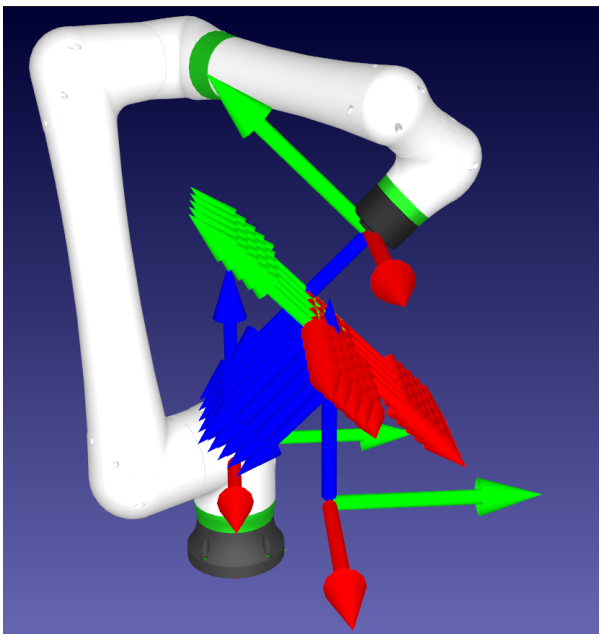


Figure 4.11: Probe offset in simulation.

4.6. Selective Target Removal

In some cases, the complexity of the grid may be such that no valid path exists without encountering jumps or exceeding cable limits. In these scenarios, it may not be feasible to include every point in the scan. However, often only a subset of the grid is sufficient to capture the most significant data. To handle this, the algorithm is designed to discard problematic points that would otherwise cause invalid movements. It then selects the smoothest path with the highest possible number of points, thereby maximizing data quality without compromising safety.

5

Optimal Antenna Placement

5.1. Introduction

The reach of the robotic arm is limited to a maximum of 1418 mm[1]. Therefore, the AUT must be placed properly to avoid invalid solutions and ensure the feasibility of the measurement paths. Moreover, The smoothness of a path in a spherical scan is highly influenced by the placement of the AUT, as the robotic arm's joints experience greater movement when it is operating near its base, which results in an invalid solution. By studying the cost of the optimal path for different antenna positions for different spherical scan parameters (r, θ_0, ϕ), where r is the radius of the measurement grid, and θ_0 is the truncation angle of the measurement grid, the optimal placement position can be located.
The origin of the global reference system for the AUT placement study is chosen as the origin of the robotic arm, as shown in Figure 5.1. With the origin at $(0, 0, 0)$, the red, green and blue arrows indicate the positive x-, y- and z-axes, respectively, as shown in Figure 5.1. For the AUT, a local reference frame is defined at the antenna with the antenna at $(0, 0, 0)$. This is needed to correctly orient the probe of the robotic arm on the grid of points and to make sure the grid is shifted with respect to the antenna (local reference) and not with respect to the robotic arm (global reference).

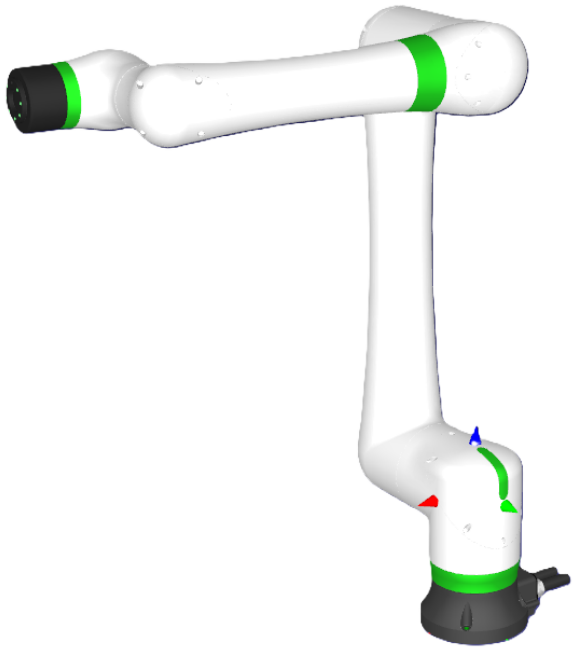


Figure 5.1: Reference of the origin of the robotic arm.

[1]CRX-20iA/L: `https://crx.fanuc.eu/wp-content/uploads/2023/12/MDS-04849-EN.20231215110957220.pdf`

5.2. Method

To obtain the cost of optimal paths for different input combinations, the robotic arm is simulated efficiently using MATLAB, and the result is validated on the physical robotic arm for several input combinations. Normally, the robotic arm is simulated using MATLAB and RoboDK. However, to speed up the computations and increase the number of input combinations for r and θ_0, only MATLAB is used for simulation. The disadvantage of not using RoboDK, in which the specific robotic arm model is imported, is not being able to detect collisions. To account for this in MATLAB, the measurement radius (r) along with the probe offset is compared with the distance from the robotic arm to the antenna. A collision occurs when the distance between the robotic arm and the antenna is less than the sum of the measurement radius, the probe offset, and an optional safety margin, as described in Equation (5.1). Note that Z_{antenna} is not taken into account, since the base of the robot can be seen as a plane in the positive z direction.

$$r + \max(\text{probe\_offset}) + \text{margin} > \sqrt{x_{\text{antenna}}^2 + y_{\text{antenna}}^2} \tag{5.1}$$

The setup of the robotic arm allows for 360° of freedom (Figure 5.2). Taking into account the limit of the robotic arm, the probe offset and the radius of measurement that add up to the antenna position, the limits of the AUT's location along the x- and y- axis are chosen as:

$$x_{\text{antenna}} = [-1000, 1000]\text{mm}$$

$$y_{\text{antenna}} = [-1000, 1000]\text{mm}$$

The AUT can be elevated using a height-adjustable table in the measurement setup at the Earl McCune XG Laboratory (Figure 5.2). The limits of the AUT's location along z-axis are chosen as:

$$z_{\text{antenna}} = [-50, 50]\text{mm}$$

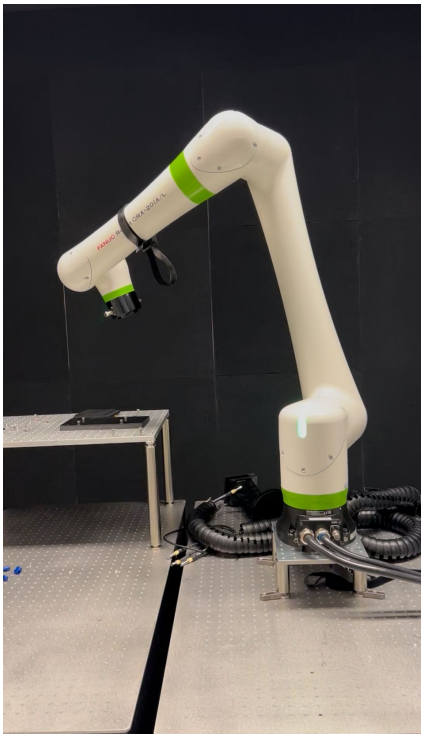


Figure 5.2: Earl McCune XG Laboratory robotic arm setup.

Considering the large size of the grid and the time-extensive calculations for each simulation, the grid is divided into 100 equidistant points. Every point (x_{antenna}, y_{antenna}) is simulated for different

z_{antenna}, different input combinations (r, θ_0) and different sorting algorithms (theta cuts and snake). For simulations, ϕ is fixed. This results in the following parameters:

$$
\begin{aligned}
x_{\text{antenna}}, y_{\text{antenna}} &\in [-1000, 1000] \text{ mm} \\
z_{\text{antenna}} &\in [-50, 50] \text{ mm} \\
r &\in \{50, 100, 150, 200, 250, 300\} \text{ mm} \\
\theta_0 &\in \left\{0 : \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{5\pi}{12}, \frac{\pi}{2}\right\} \\
\phi &= [0, 2\pi] \text{ rad} \\
\text{Probe offset} &= [0, 0, 140] \text{ mm}
\end{aligned}
$$

As a field, spherical is chosen. The polarization option is chosen as the Ludwig-III definition for a y-polarized antenna in this study (Appendix A).
In addition to the cost output, a valid flag is used to indicate whether a solution is valid or if an error occurred, which resulted in an invalid solution. When there is a collision or when a target point is out of bounds, an error is raised, resulting in an invalid flag. The simulation code can be seen in Appendix B.

5.3. Simulation Results

Figure 5.4 and Figure 5.3 illustrate an example of the minimum cost value map resulting from variations in the AUT location along the x- and y-axes, where the AUT is assumed to be located at the z=0 plane, with the spherical scan parameters set as r = 150mm, θ_0= 60°, for "snake" and "theta cuts" sorting algorithms respectively. The boxes around the point are filled with the cost-color for visual purposes.

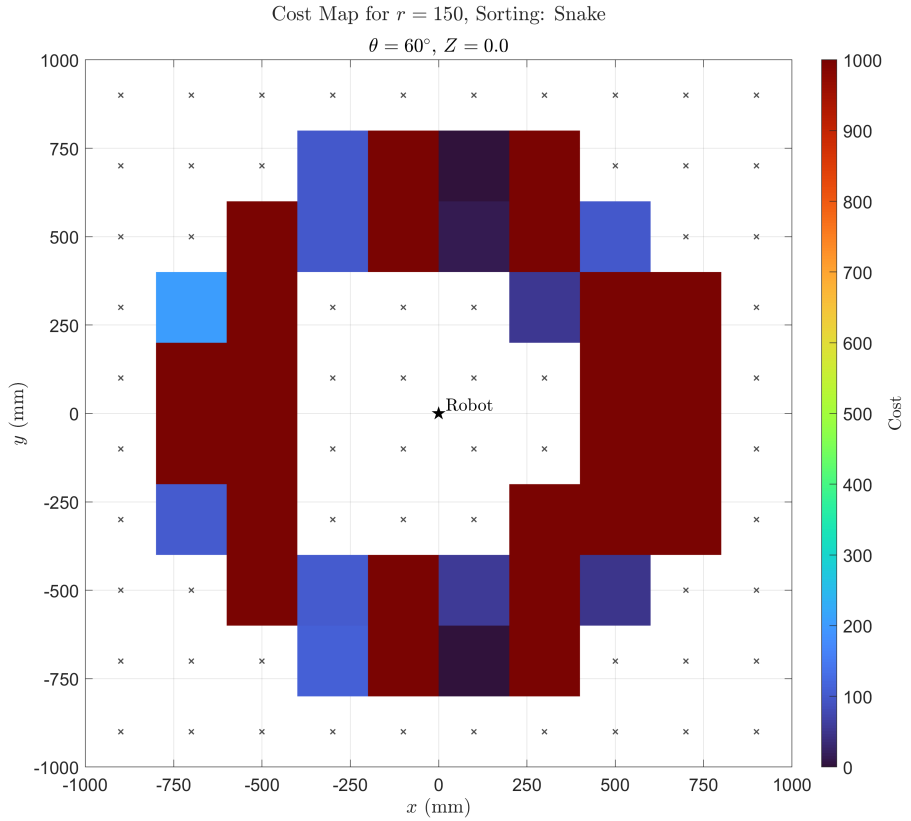


Figure 5.3: Cost map for snake with r = 150, θ = 60° and z = 0.

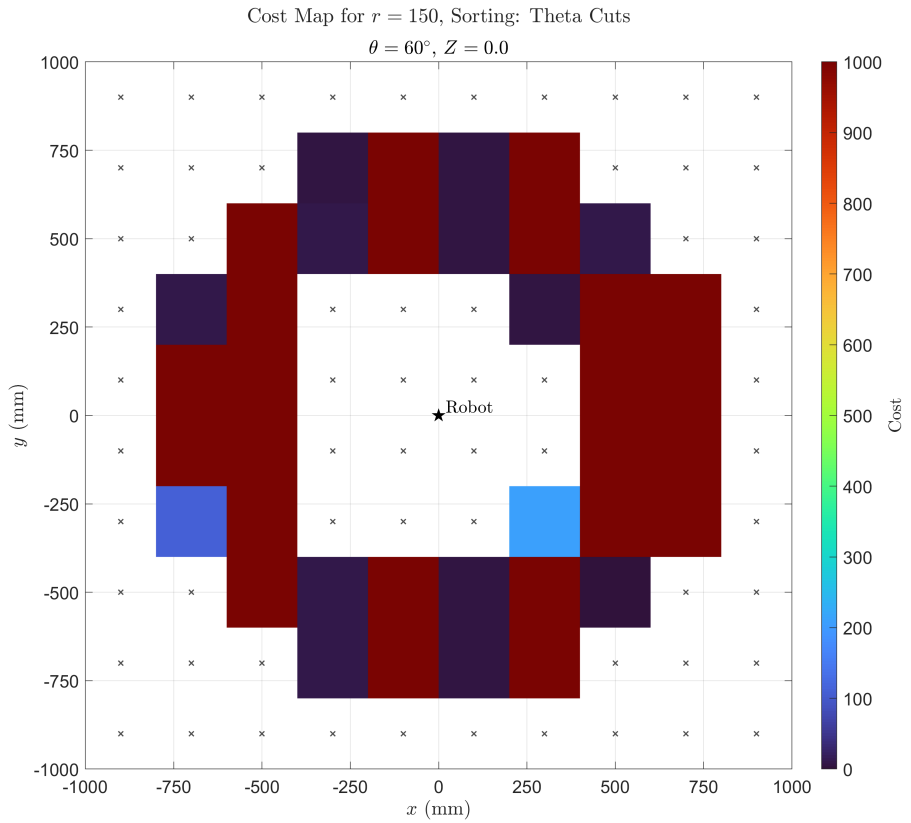


Figure 5.4: Cost map for theta cuts with $r = 150$, $\theta = 60°$ and $z = 0$.

The results for all input combinations can be found in Appendix A. It can be observed that as the AUT placement moves continuously in space, the minimum cost value changes abruptly from 1000 (red) to 0 (blue), which is counterintuitive to what one might expect. For example points $(-300, 500, 0)$ and $(100, 500, 0)$ have a sharp edge between dark red and dark blue.

To investigate the reason for the abrupt change in the joint configuration, the joint configuration (J_1-J_6) is plotted for all the targets points for $(-300, 500, 0)$ and $(100, 500, 0)$. The results can be seen in figures 5.5 and 5.6. It becomes clear in Figure 5.6, joint 6 reaches a limit at target point 70. To account for this, joint 1 and 3 have to make a jump to reach the next point. This causes the high cost (red) compared to the cost of AUT = (-300,500,0), as shown in Figure 5.5. Currently, the current path palnning algorithm tries to minimize the total joint movement from J_1 to J_6, when moving from the current pose to the next pose. This can be problematic as when J_6 reaches its limit, the algorithm tend to change J_1 J_3 abruptly to minimize the total joint movement including J_6. For future work, the joints could be weighted, to possibly decrease the number of jumps in J_1-J_3, thus decreasing the cost.

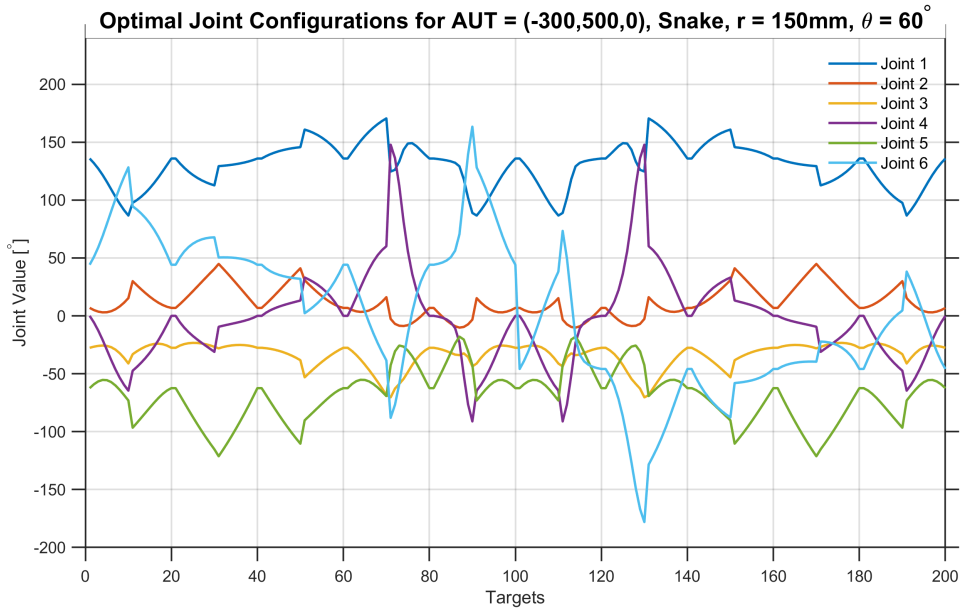


Figure 5.5: Optimal point configuration for AUT = (-300,500,0), snake, r = 150 mm and $\theta = 60°$ with a cost of 100.

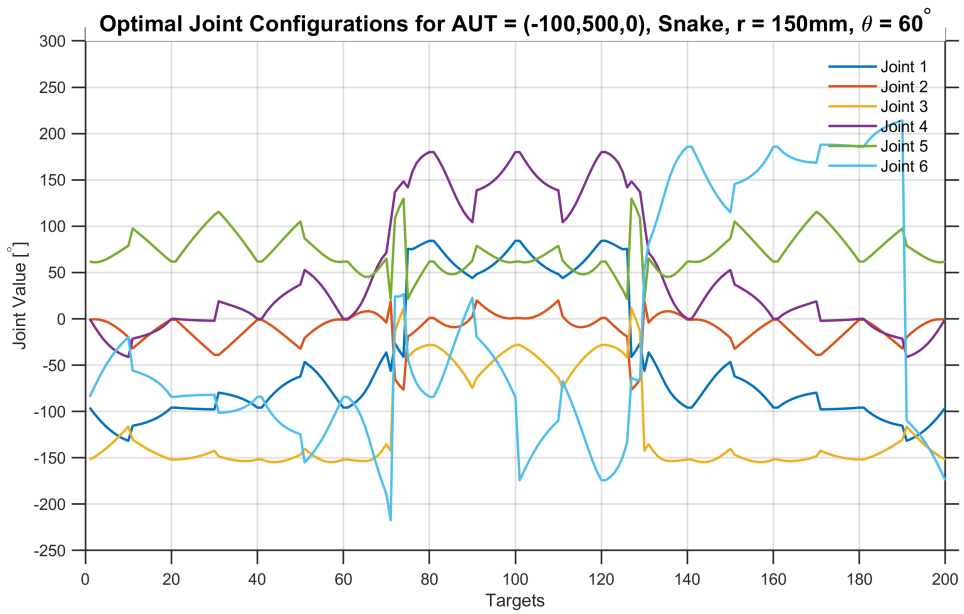


Figure 5.6: Optimal point configuration for AUT = (-300,500,0), snake, r = 150 mm and $\theta = 60°$ with a cost of 1000+.

6

Simulation and Measurements Results

6.1. Simulation Results

Combining all the simulation results from chapter 5, a look-up table for the optimal AUT placement and sorting method is created in Figure 6.1, for different scan parameters in r and θ_0 Note that this look-up table is specific for for Lugwid III definition, y-polarized antenna, and 10x10 points in θ and ϕ.

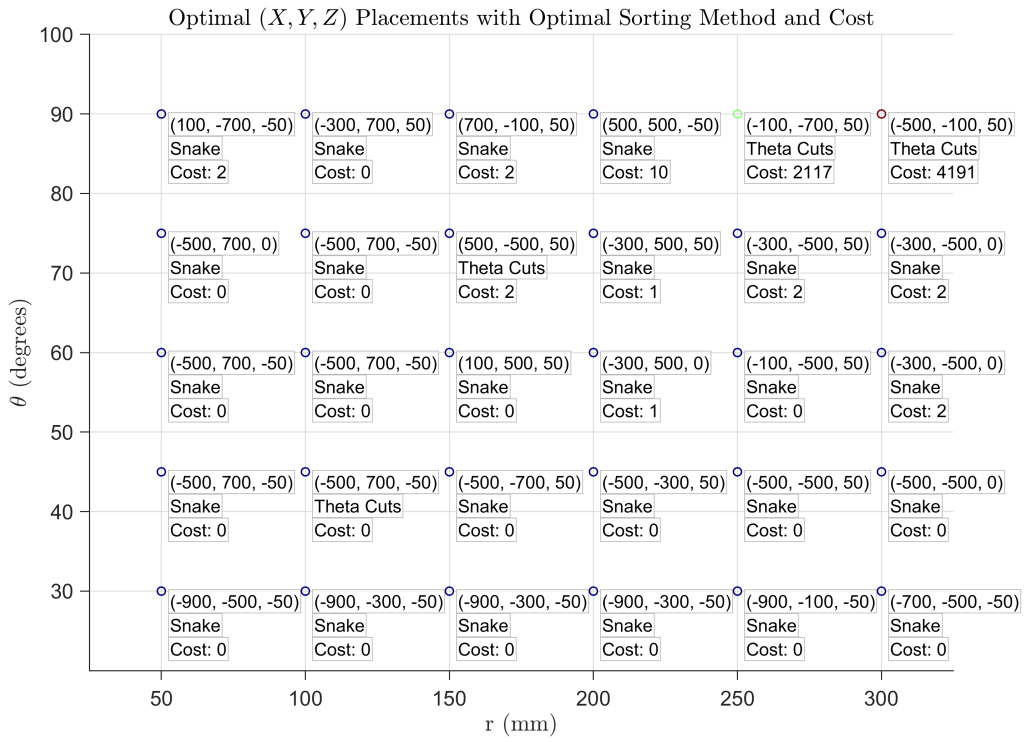


Figure 6.1: Optimal AUT placement with optimal sorting algorithm and cost.

It is observed that for most scan parameter settings, there exists an optimal AUT position where a smooth path can be achieved with a cost value of zero. However, for larger values of θ_0 ($\theta_0 > 60°$) and r ($r > 250$ mm), no smooth path is achievable.

Since only MATLAB is used and the specific robotic arm is not used via RoboDK, the obtained data must be validated with the real robotic arm. By giving the same input parameters to the robotic arm as

the simulations, the amount of jumps can be seen in the robotic arm, which translate to the cost. This is done for all input combinations (Appendix A), of which a couple of points are shown in (Table 6.1).

Table 6.1: Validation on MATLAB, RoboDK and robotic arm.

r [mm]	θ_0 [°]	AUT Position (x, y, z) [mm]	Sorting Algorithm	MATLAB Cost	RoboDK Cost	Robotic Arm Cost
150	60	(300, 500, 0)	Snake	5013	5013	x
150	60	(300, 500, 0)	Theta Cuts	4017	4017	x
150	60	(300, 300, 0)	Snake	52	52	52
150	60	(300, 300, 0)	Theta Cuts	6	6	6
50	30	(-900, -500, -50)	Snake	0	0	0
50	30	(-900, -500, -50)	Theta Cuts	2	2	2
50	90	(100, -700, -50)	Snake	2	2	2
50	90	(100, -700, -50)	Theta Cuts	56	56	56
300	90	(-500, -100, 50)	Snake	14183	14183	x
300	90	(-500, -100, 50)	Theta Cuts	4191	4191	x

Costs, that are marked in red, indicate multiple jumps in $J_1 - J_3$, which cannot be tested due the potential risk of damaging the robot or the measurement equipment. Because of this, a "x" is used for the robotic arm cost. As for the points $(300, 500, 0)$ and $(300, 300, 0)$, there is indeed a sharp edge, i.e. a sudden increase in cost, which implies that the robotic arm reaches a limit there, which happens abruptly.

6.2. Complete System Measurements Results

The complete system with the robotic arm is measured. The probe, attached to the robotic arm, has a probe offset of [0,0,145]mm. Before testing, the antenna placement was calibrated onto the robotic arm. In Table 6.2, the scan type, input parameters and number of points for x,y or θ,ϕ can be seen. The results are shown in Table 6.3, where the cablibrated antenna position, cost and orientation shift in joint six is shown.. For spherical scan, the optimal antenna placement is found using the optimal antenna placement algorithm. This position is then also calibrated onto the robotic arm.

Table 6.2: Scan definitions and input parameters.

Scan Type	Input Parameters ((x, y), Z_{offset}, Sorting Algorithm, Polarization Axis)	Number of Points
Planar	$x, y \in [-46.14, 46.15]$ mm, $z_{\text{offset}} = 32.5$ mm, "Snake long", "Y-axis"	625 (25 x 25)
Tilted Planar	$x, y \in [-46.14, 46.15]$ mm, $z_{\text{offset}} = 32.5$ mm, $\theta = 16°$, "Snake long", "Y-axis"	625 (25 x 25)
Spherical	$r = 174$ mm, $\theta = 60°$, "Snake", "Y-axis"	1089 (33 x 33)

Table 6.3: System measurement results.

Scan Type	Antenna Position (x, y, z) [mm]	Cost	ΔJoint 6 [°]
Planar	[452.48, 753.10, −243.10]	0	98.13
Tilted Planar	[499.97, 760.60, -146.26]	0	101.11
Spherical	[398.83, 699.84, 59.30]	0	245.03

7

Conclusion and Future Work

This thesis presented an automated system for antenna measurements in both NF and FF using a 6 DoF Fanuc CRX-20iA/L robotic arm. A path planning algorithm was developed that minimizes joint discontinuities ("jumps") by evaluating paths through a weighted cost function. An IK method was used to obtain all joint configurations, allowing selection of the smoothest, least strained path for any given measurement grid.

Simulation and experimental validation confirmed that the proposed method successfully avoided jumps in all six joints for both planar (NF) and spherical (FF) configurations. A cost of zero was achieved in both cases, meeting the requirements of all robotic arm subsystem requirements (B-F.1 to B-F.8, B-NF.1). Additionally, both Ludwig II and III polarization configurations were supported for spherical measurements (B-F.3).

Antenna placement optimization significantly impacted the minimal cost of the path. By simulating over 100 placement positions and evaluating them for various input combinations of measurement radius and angular limit, the optimal AUT positions were created to ensure path smoothness, minimize cost and eliminate invalid paths due to reach limits or cable strain. These simulations were validated against a commercial robotic arm simulation software RoboDK and real-world measurements, showing consistent results.

Furthermore, the software was implemented in MATLAB with a modular, object-oriented design, allowing for flexible configuration and integration with the control interface. Sorting strategies such as Snake and Theta Cuts were shown to be effective across different input configurations, with the system selecting the path with the lowest cost.

Table 7.1: Robotic arm requirements checklist.

ID	Requirement Description	Accepted
B-F.1	Uses X,Y,Z ($N \times 3$) coordinates from subgroup 4 for planar (NF) scans	Yes
B-F.2	Uses r, θ, ϕ ($N \times 3$) coordinates from subgroup 3 for spherical (FF) scans	Yes
B-F.3	Supports spherical scans using Ludwig II and Ludwig III polarization definitions	Yes
B-F.4	Follows the smoothest possible path	Yes
B-F.5	Performs antenna pattern measurements with angular resolution of 1° or finer	Yes
B-F.6	Receives 'move-to' commands from subgroup 2 as joint-configuration vectors	Yes
B-F.7	Sends digital readiness signal (logic HIGH) within 0.1 mm tolerance	Yes
B-F.8	Allows adjustable motion speeds	Yes
B-NF.1	Control algorithms implemented in MATLAB	Yes
B-NF.2	Software is modular to allow future upgrades with minimal redesign	Yes

Future work will explore alternative sorting algorithms to enhance the performance of grid sorting. Additionally, extending the optimization algorithm to support more input configurations and account for the number of points per configuration could lead to more efficient measurement setups. Finally, cable routing should be further investigated, as improper placement can compromise path validity by increasing the risk of cable strain. Moreover, there is a risk of a jump when the robotic arm moves to the first point, which can be taken into consideration. The path could be smooth, but the path towards the first point may not be. Also a weight could be added to the joints, to decrease the number of jumps in J_1-J_3, as discussed in chapter 5.

Bibliography

[1] C. A. Balanis, *Antenna theory: analysis and design*, 3. ed. Hoboken, N.J, 2005.

[2] M. H. Francis and R. C. Wittmann, "Near-field scanning measurements: Theory and practice," in *Modern Antenna Handbook*, 1st ed., 2008.

[3] P. Pinho, H. Santos, and H. Salgado, "Design of an anechoic chamber for w-band and mmWave," *Electronics*, vol. 9, no. 5, p. 804, 2020.

[4] A. Husein, K. Rasilainen, J.-P. Mäkelä, A. Parssinen, and M. E. Leinonen, "6g ota measurements at sub-thz band using a compact robotic system," presented at the 2025 19th European Conference on Antennas and Propagation (EuCAP 2025), Stockholm, Sweden, 2025.

[5] L. Boehm, S. Pledl, F. Boegelsack, M. Hitzler, and C. Waldschmidt, "Robotically controlled directivity and gain measurements of integrated antennas at 280 GHz," presented at the 2015 European Microwave Conference (EuMC 2015), Paris, France, 2015.

[6] J. A. Gordon, D. R. Novotny, M. H. Francis, *et al.*, "Millimeter-wave near-field measurements using coordinated robotics," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 12, 2015.

[7] C. Weidemann, N. Mandischer, F. Van Kerkom, *et al.*, "Literature review on recent trends and perspectives of collaborative robotics in work 4.0," *Robotics*, vol. 12, no. 3, 2023.

[8] M. Abbes, K. Belharet, M. Souissi, H. Mekki, and G. Poisson, "Design of a robotized magnetic platform for targeted drug delivery in the cochlea," *IRBM*, vol. 44, no. 1, Feb. 2023.

[9] M. Abbes, K. Belharet, H. Mekki, and G. Poisson, "Use of the CRX10ia cobot for microparticles delivery inside the cochlea," presented at the 2023 Conference on New Technologies for Computer and Robot Assisted Surger, Paris, France, 2023.

[10] S. F. Gregson and C. G. Parini, "Use of compressive sensing techniques for the rapid production test of commercial nose-mounted radomes in a robotic antenna measurement system," 2025.

[11] B. L. Moser, J. A. Gordon, and A. J. Petruska, "Increased efficiency in planar near-field scanning using combined multi-robot motion," presented at the 2025 19th European Conference on Antennas and Propagation (EuCAP 2025), Stockholm, Sweden, 2025.

[12] K. Dausien, T. Korner, C. Schulz, N. Pohl, I. Rolfes, and J. Barowski, "Ultrawideband millimeterwave robotic antenna measurements enabled by FMCW radar sensors," 2025.

[13] H. Jansen, R. Moch, and D. Heberling, "Electrical alignment technique for offset-mounted and arbitrarily oriented AUTs in a robot-based mm-wave antenna test system," in *2023 Antenna Measurement Techniques Association Symposium (AMTA)*, Renton, WA, USA, Oct. 8, 2023.

[14] N. A. Aboserwal, J. L. Salazar, J. A. Ortiz, J. D. Diaz, C. Fulton, and R. D. Palmer, "Source current polarization impact on the cross-polarization definition of practical antenna elements: Theory and applications," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 9, 2018.

[15] *IEEE standard for definitions of terms for antennas*, New York, 2013.

[16] A. Ludwig, "The definition of cross polarization," *IEEE Transactions on Antennas and Propagation*, vol. 21, no. 1, 1973.

[17] D. L. Pieper, "THE KINEMATICS OF MANIPULATORS UNDER COMPUTER CONTROL," 1968.

[18] M. Abbes and G. Poisson, "Geometric approach for inverse kinematics of the FANUC CRX collaborative robot," *Robotics*, vol. 13, no. 6, 2024.

[19] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 1986.

[20] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, Jun. 1, 1955.

[21] J. Denavit, R. S. Hartenberg, R. Razi, and J. J. Uicker, "Velocity, acceleration, and static-force analyses of spatial linkages," *Journal of Applied Mechanics*, vol. 32, no. 4, 1965.

[22] L. Carbonari, M.-C. Palpacelli, and M. Callegari, "Inverse kinematics of a class of 6r collaborative robots with non-spherical wrist," *Robotics*, vol. 12, no. 2, 2023.

[23] X. Tang, H. Zhou, and T. Xu, "Obstacle avoidance path planning of 6-DOF robotic arm based on improved a* algorithm and artificial potential field method," *Robotica*, vol. 42, no. 2, 2024.

[24] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, 2010.

A

Antenna Optimization

In this appendix the simulation results for all input combinations of θ and ϕ are shown for the snake and theta cuts sorting algorithms for 100 points.

Snake

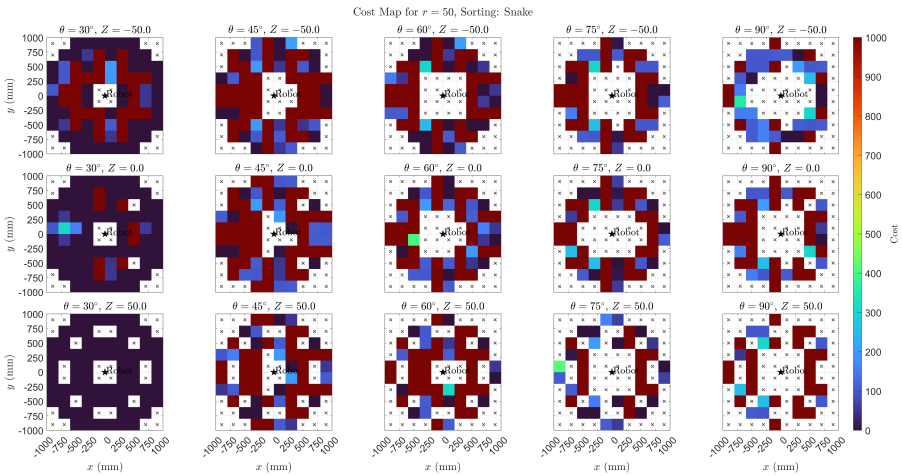


Figure A.1: Cost map for snake with $r = 50$.

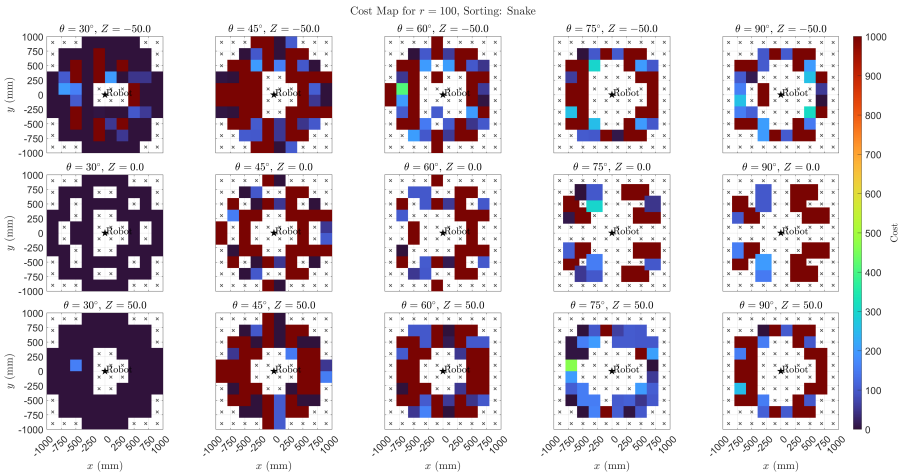


Figure A.2: Cost map for snake with $r = 100$.

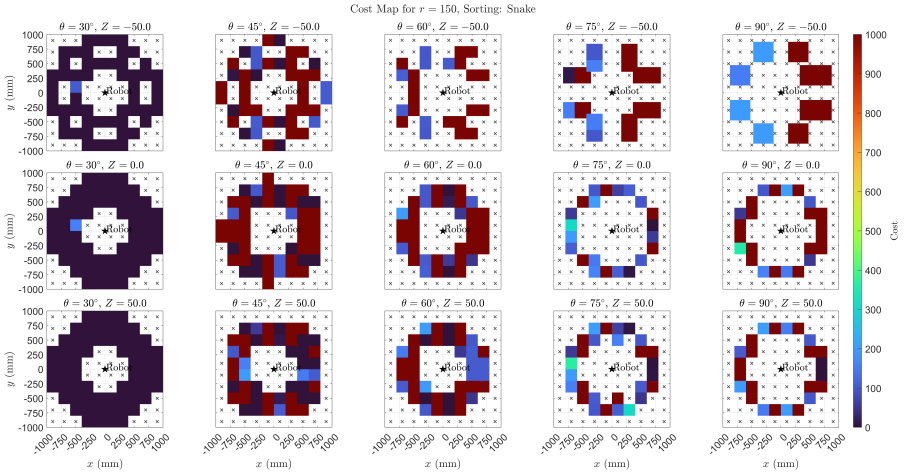


Figure A.3: Cost map for snake with $r = 150$.

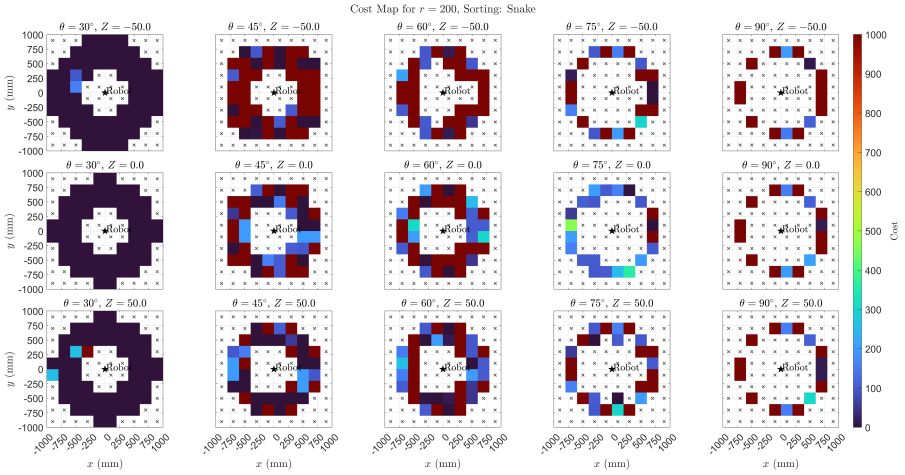


Figure A.4: Cost map for snake with $r = 200$.

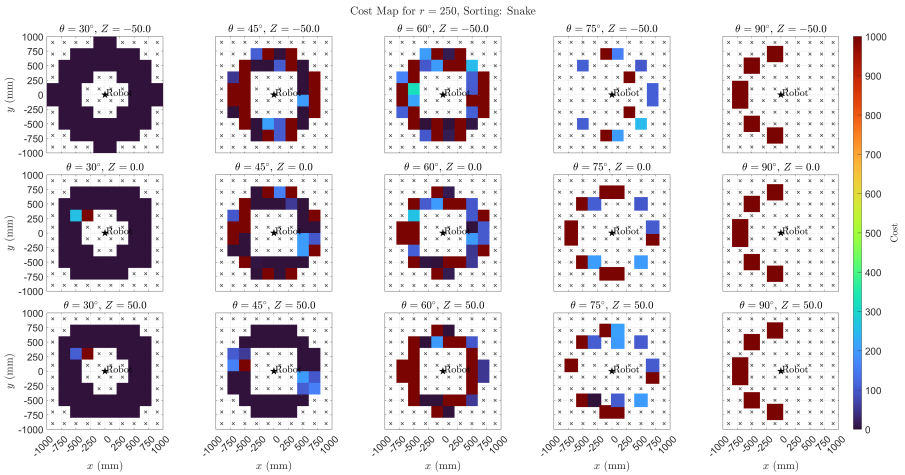


Figure A.5: Cost map for snake with $r = 250$.

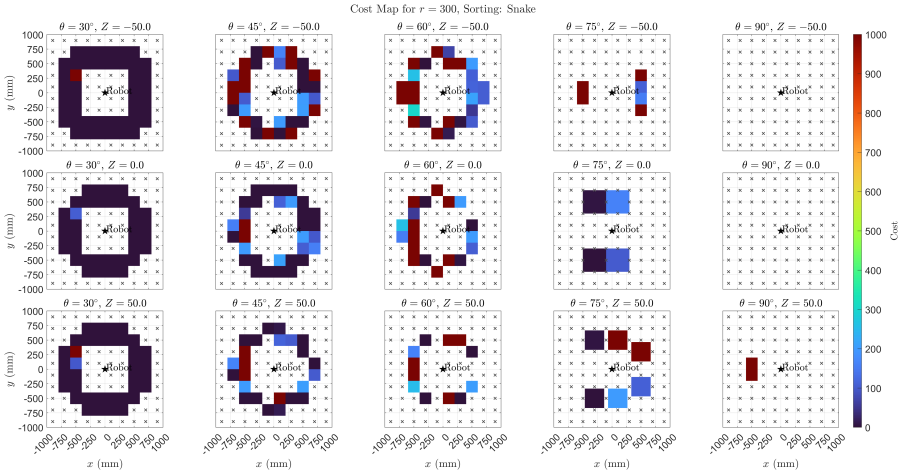


Figure A.6: Cost map for snake with $r = 300$.

Theta cuts

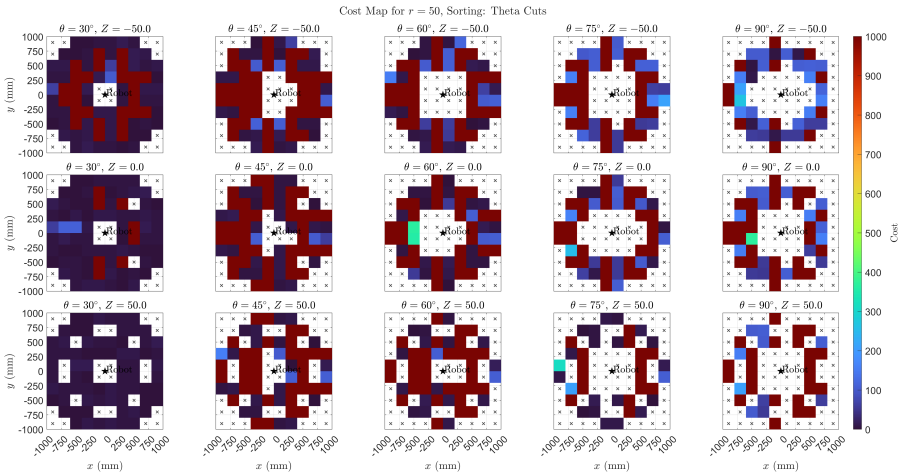


Figure A.7: Cost map for theta cuts with $r = 50$.

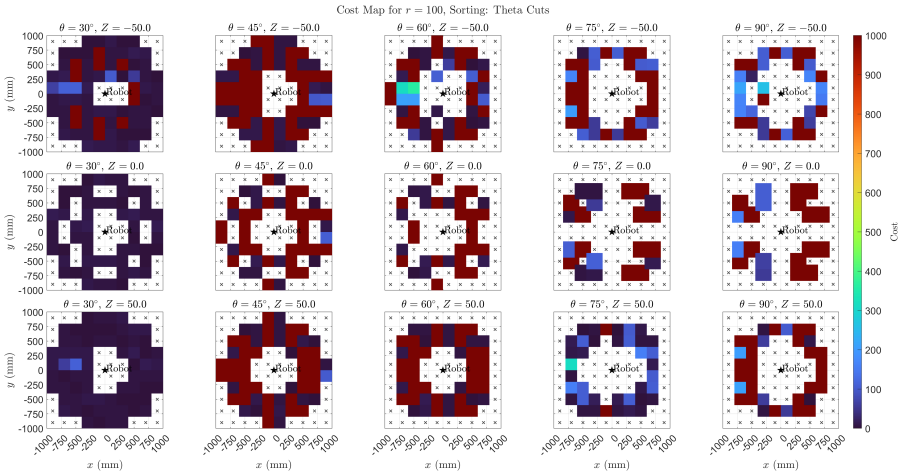


Figure A.8: Cost map for theta cuts with $r = 100$.

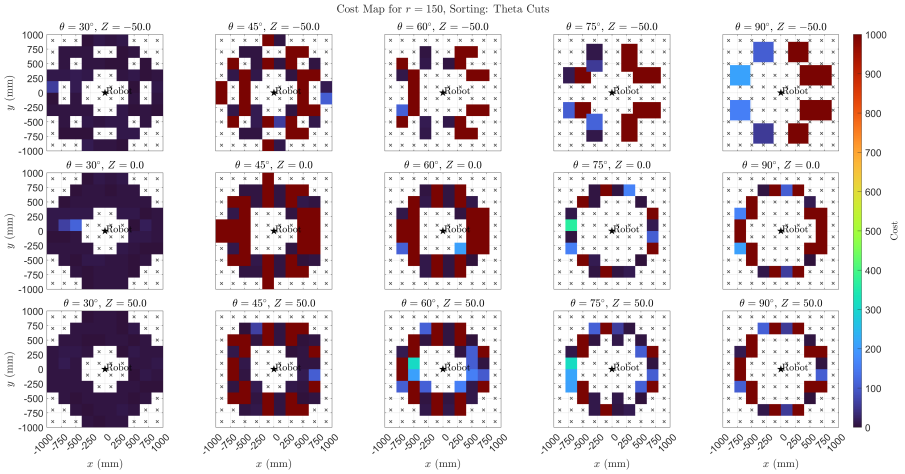


Figure A.9: Cost map for theta cuts with $r = 150$.

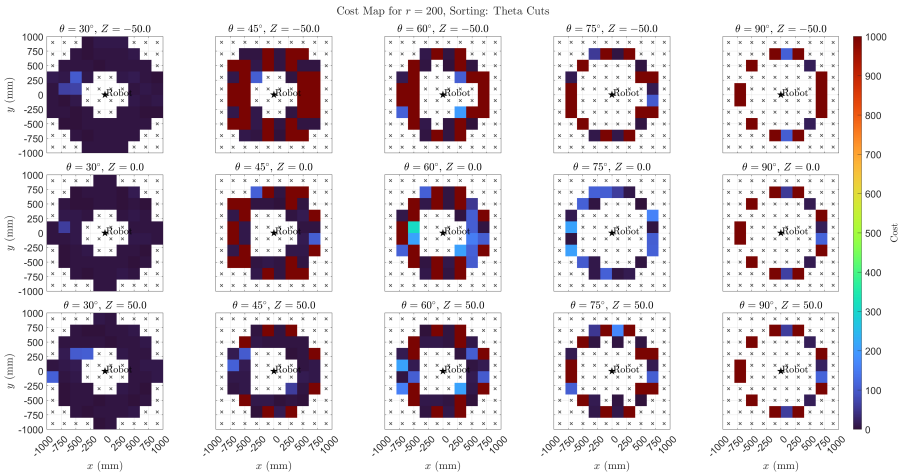


Figure A.10: Cost map for theta cuts with $r = 200$.

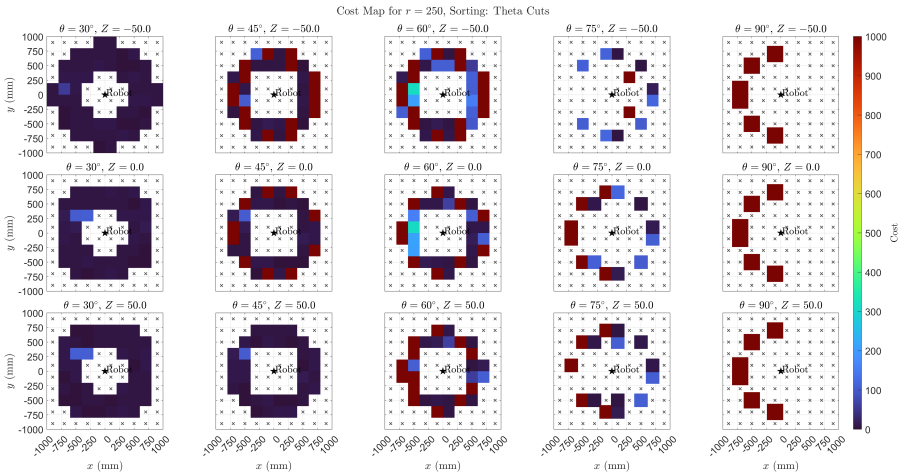


Figure A.11: Cost map for theta cuts with $r = 250$.

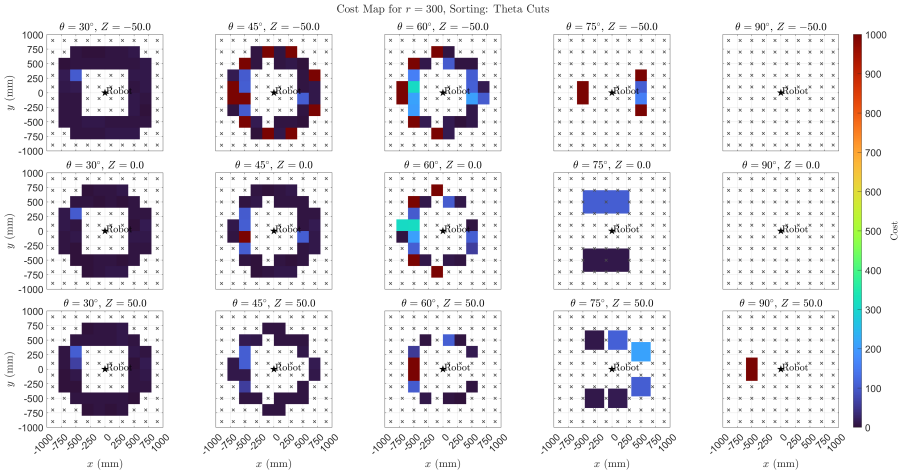


Figure A.12: Cost map for theta cuts with $r = 300$.

B

MATLAB code

B.1. Robot Class

To increase the modularity of the code for the robotic arm and to increase the ease of use for the control interface (subgroup 2), an Object-Oriented approach is taken. By programming in classes, the inputs and outputs of functions can be stored in a robot class, compared to storing them in the interface. By doing this, the interface only has to call the function name, after which the class gives the right inputs. For example, in the code below, a path is created. In $main.m$ only the create path is called with the grid input from subgroup 3 or 4. In the robot class ($Robot.m$), the other inputs are given.

```
%%Main.m
robot_instance.create_path(Grid);

%%Robot.m
function create_path(self, Grid)
    [self.Targets, self.Polarizations] = Create_targets(self.RDK, self.ref, self.robot_arm,
    ↪  self.field, Grid, self.Ludwig, self.theta_planar, self.phi_planar, self.sorting_way);
    [self.joint_config, self.cost_best] = Generate_path(self.robot_arm, self.Tcp_probe,
    ↪  self.ref, self.Targets);
end
```

Moreover, this approach makes the robot class modular and easy to use. Since the control interface ($Main.m$) does not need to know what the inputs are specifically, the functionalities of the robot class can be changed without having to change the entire control interface.

Another benefit of using an OO-approach is the protection of variables connected to the robot class. In the robot class, there are private and public variables. Private variables can only be accessed from within the robot class, while public variables can be accessed from the controller interface. The input variables of the user, such as the range of θ, can be controlled from within the controller interface, while more software-critical parameters, such as the reference of the robotic arm, are locked into private variables.

```
classdef Robot < handle
    properties (Access = private)
        Tcp_probe = eye(4);
        RDK;
        station;
        robot_arm;
        select_mode;  % 0 for simulation, 1 for robot and simulation
        filename_rdk = 'CRX.rdk';
        filename_robot = 'Fanuc-CRX-20iA-L.robot';
        ref_Name = 'Reference Frame Measurement Field';
        ref;
        Distance_antenna = [-300,500,0];
        Offset_antenna = [0,0,140];
```

```
        LinearSpeed = 100;
        JointSpeed = 50;
    end
    properties (Access = public)
        Targets;
        Polarizations;
        joint_config;
        cost_best;
        field = 0;
        Grid_given = 0; % 0 make grid, 1 grid given
        polaxis = 'Y'; % polarization axis of the source
        % (Tilted) Planar:
        x_range = -100:50:100;
        y_range = -100:50:100;
        z_offset = 250;
        theta_planar = pi/2; % Rotation on theta
        phi_planar = pi/2; % Rotation on phi
        % Spherical:
        Ludwig = 3; % 0 for Ludwig 2, 1 for Ludwig 3
        sorting_way = 'theta_cuts';
        theta_range = deg2rad(-50:20:50);
        phi_range = deg2rad(-80:40:80);
        r = 200; % radius of measurement
    end
```

However, private parameters can be controlled and changed with the so-called getter and setter functions. These functions provide a controlled way of changing the parameters:

```
function set_linear_speed(self,linear_speed)
    self.LinearSpeed = linear_speed;
    self.robot_arm.setSpeed(linear_speed, self.JointSpeed);
end

function linear_speed = get_linear_speed(self)
    linear_speed = self.LinearSpeed;
end
```

These functions can be called within the controller interface, because of which the private parameters could be changed if necessary.

In this Github link, the code used for controlling the robotic arm, generating a path, and following a path is shown:
`https://github.com/hawmaas/BEP_group1.git`
This Github repository contains the following files:

- Main.m
- Robot.m
- create_grid.m
- grid_sort.m
- Create_targets.m
- FindPose.m
- FindPose_Planar.m
- rotate.m
- Generate_path.m
- find_jump.m
- plot_derivative_peaks.m
- Follow_path.m
- Main_Antenna.m
- get_current_position.m
- move_robot.m
- IK_FanucCRX.m
- O34_dot_O45_ZeroFind.m
- O34_dot_O45.m
- T_DHm.m