

Methodical Approach for Analyzing Process Variables and Optimizing Boundary Conditions in Multi-Axis Robot Programs

Methodischer Ansatz zur Analyse von Prozessvariablen und Optimierung von Randbedingungen in Multi-Achs-Roboterprogrammen

Scientific work for obtaining the academic degree

Master of Science (M.Sc.)

at the TUM School of Engineering and Design of the Technical University of Munich

Supervised by	Prof. Dr.-Ing. Michael Zäh Institute for Machine Tools and Industrial Management (iwb)
Submitted by	Jan Nalivaika Lerchenauerstrasse 10 80809 Munich
Submitted on	March 01, 2024 in Garching

Chapter 1

Implementation and Validation

The first step in the validation process involves selecting a manufacturing machine and constructing a fundamental model that captures the kinematics and dynamics. In this case, a simple articulated robot is chosen as the manufacturing machine. To test the method in a straightforward scenario, initial tests are conducted on a 6-DoF model with a 5-DoF toolpath. The sixth DoF, which represents rotation around the Z-axis, is freely defined and utilized for optimization purposes. Once this simple model is successfully validated, an additional redundant DoF is introduced by incorporating the tilt of a rotary-tilt table.

After establishing the basic mathematical model, an optimization algorithm is implemented to determine the optimal values for each redundant DoF in order to optimize the process variables. This optimization algorithm takes into consideration the specific constraints and objectives of the industrial robot, such as minimizing direction changes or joint accelerations.

1.1 Simple Implementation

1.1.1 Modeling a 6-DoF Robot

To test the proposed method, a simplified articulated industrial robot with 6-DoF is utilized as a model. A visual representation of this robot can be seen in Figure 1.2. The Denavit-Hartenberg (DH) parameters for this robot are provided in Table 1.1. These parameters are crucial for describing the geometry and kinematics of the robotic arm. They establish the relationship between adjacent links in the kinematic chain of the robot.

The DH parameters consist of various values, including link lengths, link twists, and link offsets. In this particular model, a represents the link lengths between adjacent joints, α represents the link twists or rotations around the Z-axis between adjacent joints, and d represents the link offsets or distances along the Z-axis between adjacent joints. The units of a and d are set in millimeters, while the rotation α is defined in degrees.

Parameters	Values
a in [mm]	[200, 800, 150, 0, 0, 0]
alpha in [°]	[90, 0, 90, -90, 90, 0]
d in [mm]	[400 0, 0, 600, 0, 200]

Table 1.1: DH-parameters for the modeled robot

The schematic of the modeled robot can be observed in Figure 1.1. In this specific configuration, all joints are in their initial positions with no rotation applied.

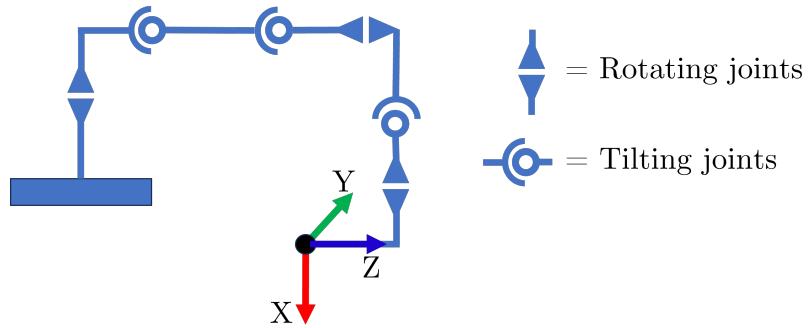


Figure 1.1: Schematics of the modeled robot

The next step is to model a spindle that is attached to the flange, which is at the end of the last link and apply a transformation matrix from the flange to the tip of the end-mill. Figure 1.2 depicts the robot modeled using Python in combination with the *Matplotlib* library. The joint positions, in degrees, are as follows: [0, 135, -45, 0, 0, 0], corresponding to joints 1 to 6, respectively. The colored arrows shown in the figure represent the coordinate axes of the TCP. The X-axis of the TCP coordinate system is represented in red, while the Y-axis and Z-axis are denoted by the colors green and blue, respectively. To distinguish the TCP coordinate system from the world coordinates system, the TCP coordinates system axes are marked with a dash. The first link of the robot, originating from the point X=0, Y=0, Z=0 in the world coordinate system, is displayed in green. The individual joints are represented by red dots. The end-mill is marked as a orange dot.

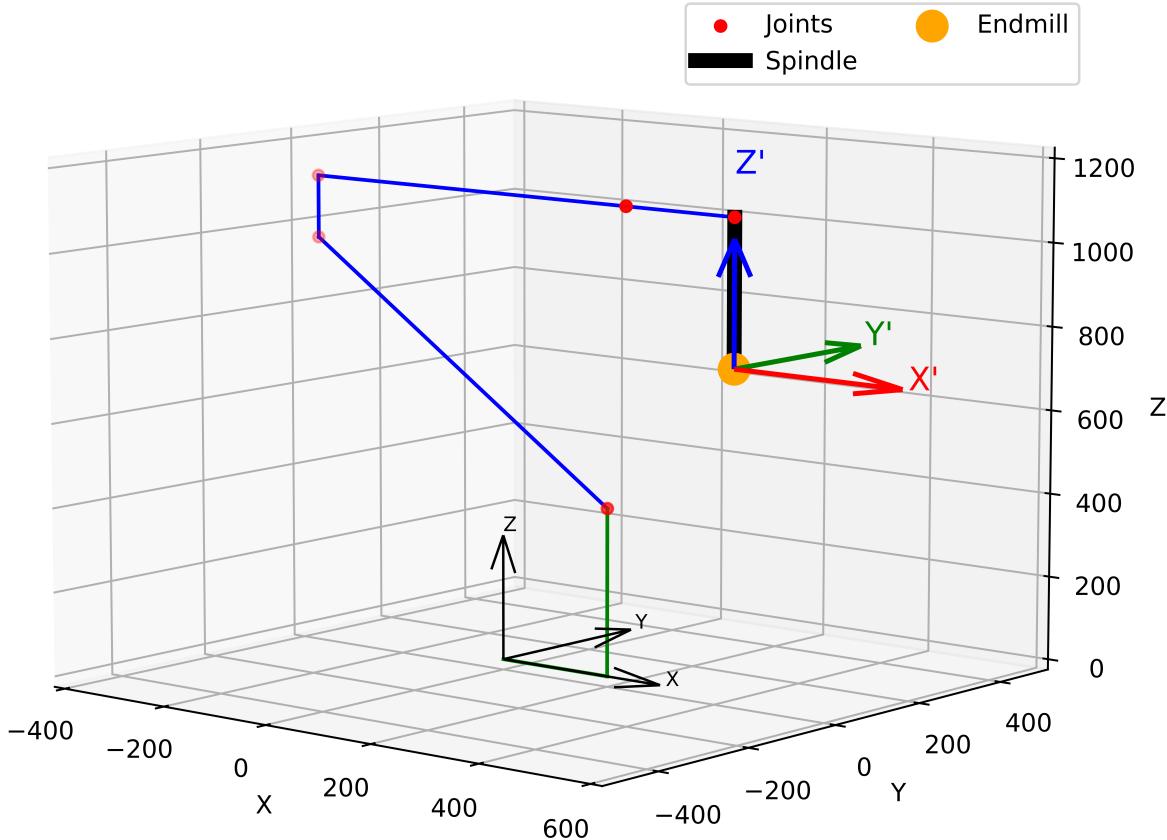


Figure 1.2: Visualization of the modeled robot in Python

1.1.2 Modeling a Basic Toolpath

Before analyzing the process variables, it is necessary to define a toolpath for the TCP to follow. For that case, three options are presented, each consisting of 3000 coordinates. It should be noted that the redundant DoF in these cases is the rotation around the Z-axis. This rotation will be adjusted to determine the optimal value for the desired outcome. A and B are held at 0°.

The first toolpath, depicted in Figure 1.3, represents a converging spiral that is shifting to the side. Figure 1.4 illustrates a converging-diverging infinity-loop, and Figure 1.5 displays a forward-moving sinusoidal curve that is following a parabolic profile. Each path consists out of 3000 coordinates.

Figure 1.6 illustrates the robot and Toolpath 1 at the final position of the toolpath. The origin of the toolpath is shifted by $X=+800$ and $Z=+400$ relative to the world coordinate system. No rotations are applied around the X, Y, and Z axes, resulting in A, B, and C being zero. As a result, the coordinate axes of the TCP are parallel to the axes of the world coordinate system.

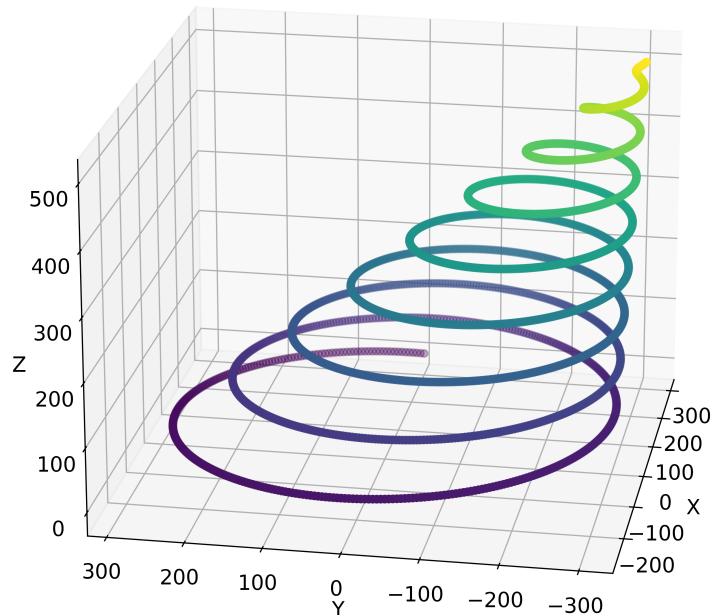


Figure 1.3: Toolpath 1: Converging spiral

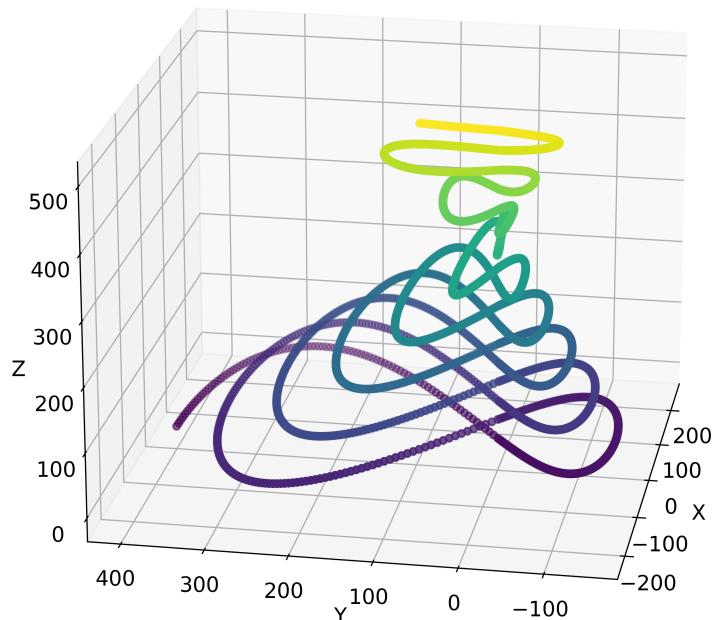


Figure 1.4: Toolpath 2: Converging infinity loop

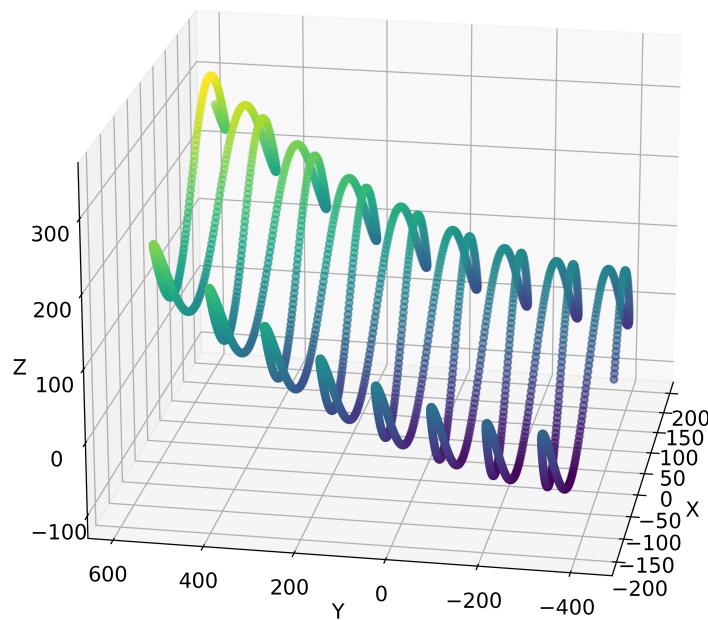


Figure 1.5: Toolpath 3: Forward moving pendulum oscillation

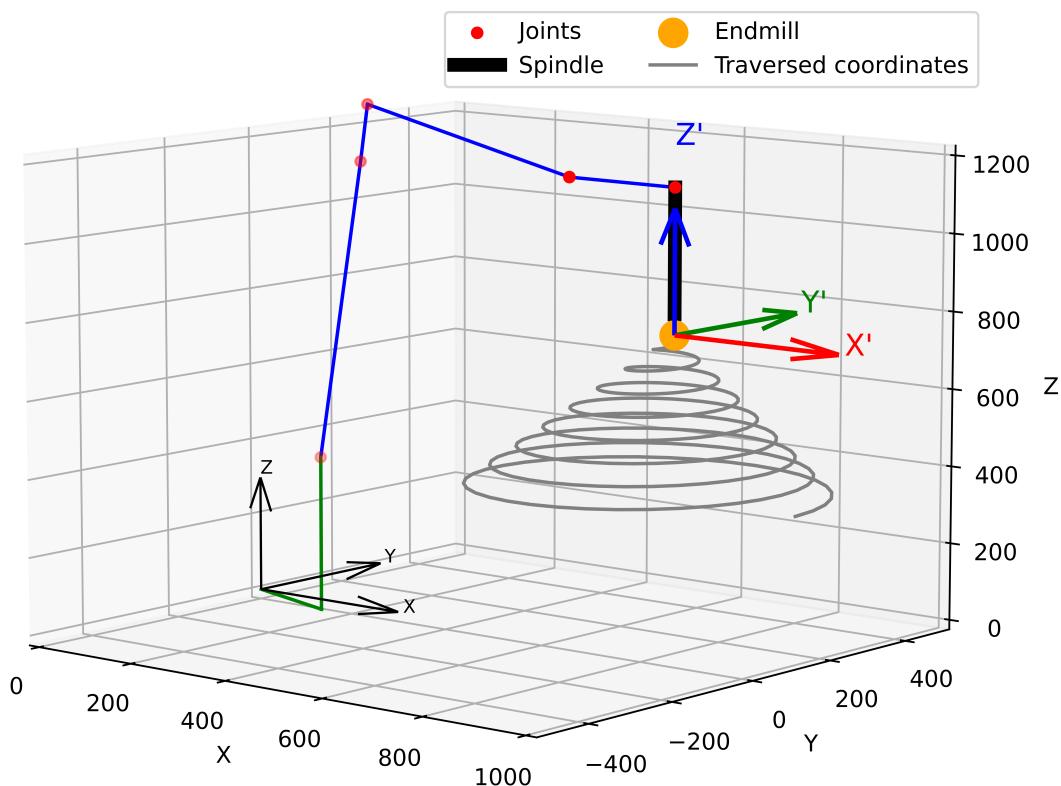


Figure 1.6: Traversing toolpath 1

1.1.3 Extracting Process Variables

By utilizing the inverse kinematics algorithm from the Python library *visual_kinematics*, the joint angles for each coordinate can be computed. To achieve this, the rotations A, B, and C need to be defined. The outcome is a time-series that contains the corresponding joint positions. For all tests, all coordinates must be traversed in equidistant time steps. With this information, it becomes possible to calculate the velocity and other related variables. By transforming all the data from the time-series into scalar values and calculating the local rating, the local and global scores can be determined.

1.2 Testing and Validation

1.2.1 Toolpath Evaluation With one Redundant DoF

As discussed in Chapter 1.1.2, the toolpath remains constant with respect to the X-, Y-, and Z-coordinates. The fixed boundary conditions for the robot are that there are no rotations around the X and Y axes, resulting in A and B both being equal to zero. This condition is fixed for the entire toolpath. The user has the ability to set the DoF for the rotation around the Z-axis, which is the redundant DoF. Figure 1.7 displays the variation of each joint over time for toolpath 1. In this specific case, the rotations A, B, and C are all set to 0. The entire toolpath is traversed in 300 seconds.

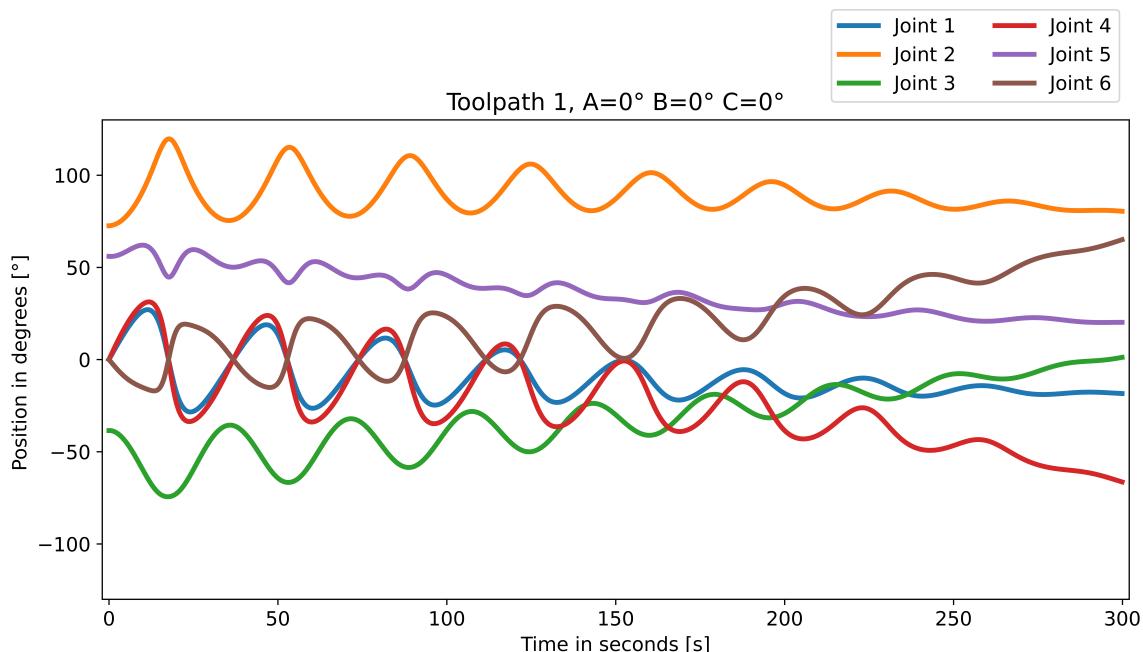


Figure 1.7: Visualization of the joint positions over time for toolpath 1 with C=0°

Figure 1.8 depicts the joint positions for all six joints over time for the same toolpath (toolpath 1) with a 45° rotation around the Z-axis ($C = 45^\circ$).

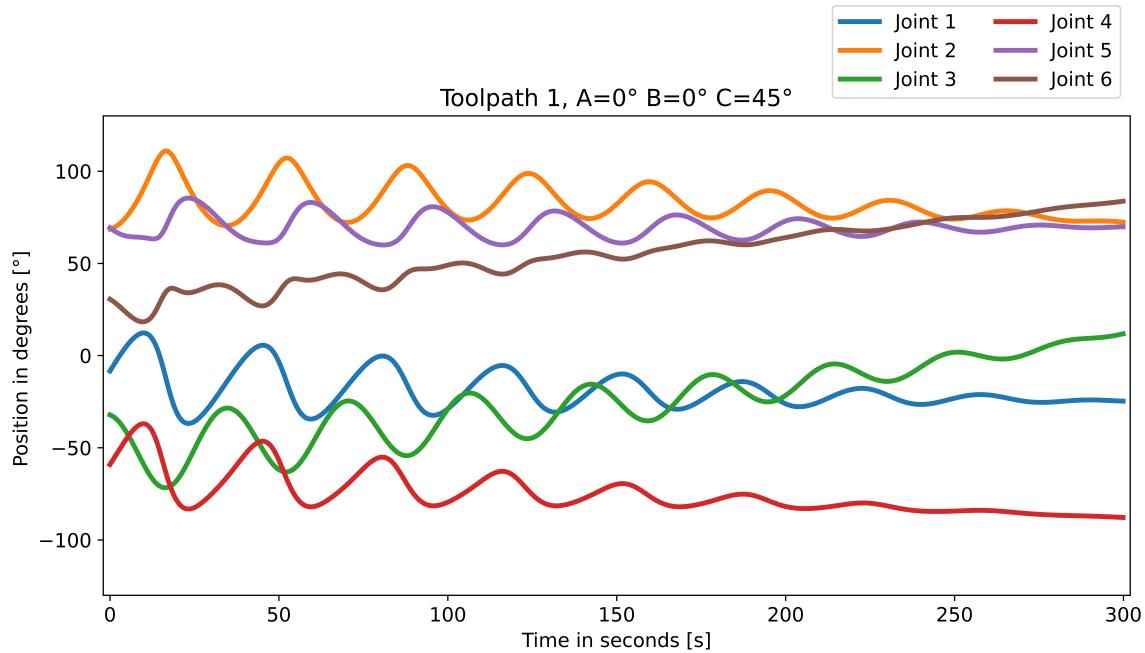


Figure 1.8: Visualization of the joint positions over time for toolpath 1 with $C=45^\circ$

It is noticeable that joint 4 and joint 5 have experienced changes in their respective ranges. Furthermore, joint 6 exhibits a significantly larger amplitude towards the end of the toolpath, in comparison to the case with no rotation ($C=0^\circ$).

Figure 1.7 illustrates the variations in each joint over time for toolpath 2 (Eq. ??) without any rotation ($A=B=C=0^\circ$). Unlike toolpath 1, the amplitudes notably decrease towards the end of the toolpath. This observation aligns with the unique characteristics of different toolpaths.

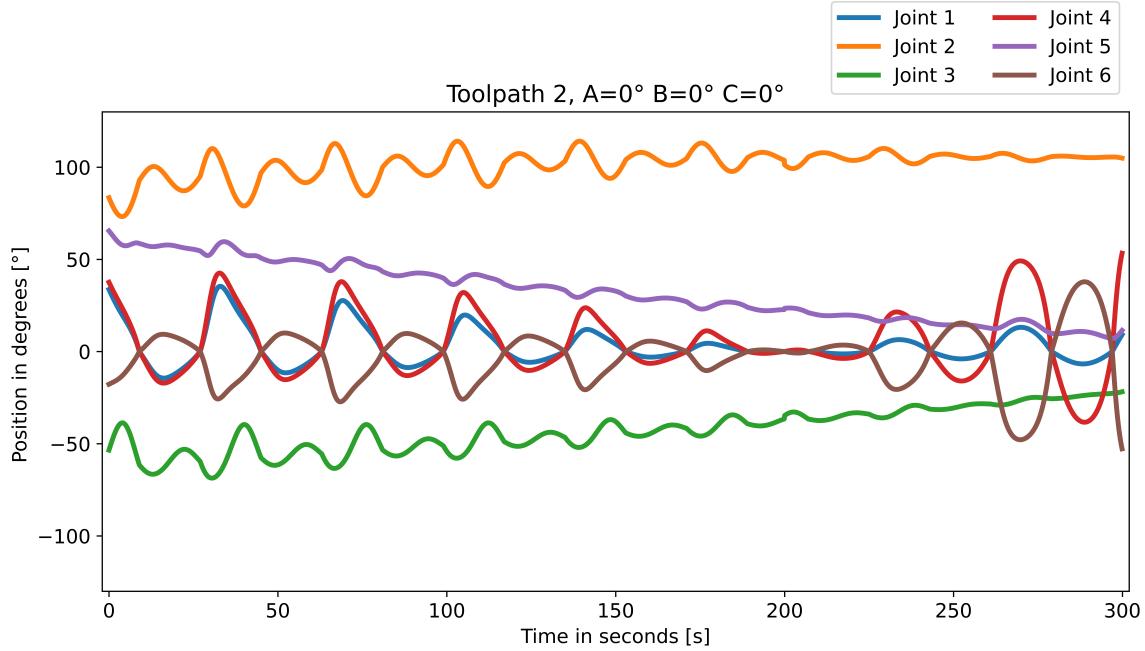


Figure 1.9: Visualization of the joint positions over time for toolpath 2

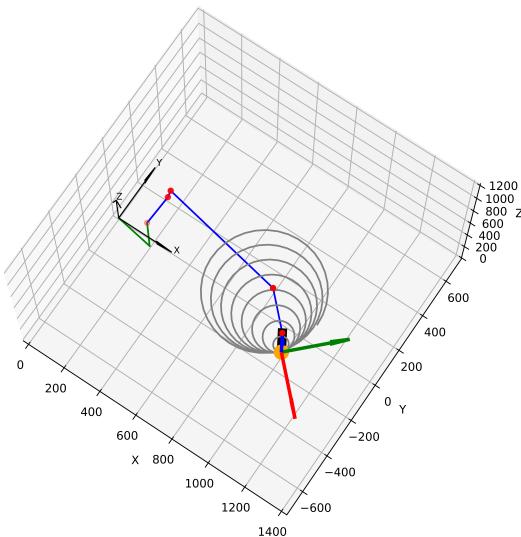
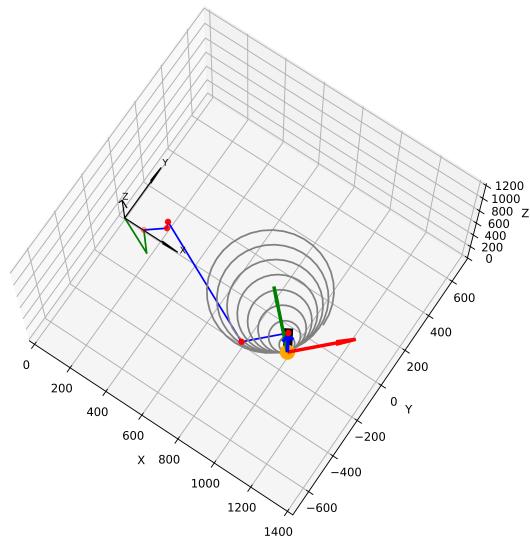
The next step involves selecting the process variables of interest and assigning weights to each variable. For this purpose, a basic case is discussed. The selected process variables are listed in Table 1.2. The total number of direction changes in all joints and the total distance traveled are chosen due to their ease of implementation. The number of direction changes is assigned an importance factor of 0.2. The total travel in all joints is combined and given an importance factor of 0.4. Additionally, the acceleration of joint 1 is analyzed. To obtain a scalar value for the acceleration, the individual acceleration values are squared and then summed up. The importance factor for acceleration in joint 1 is 0.4. Other process variables are disregarded.

Process variables	Importance factors
Direction changes in joints 1-6	0.2
Total travel in joints 1-6	0.4
Acceleration in joint 1	0.4

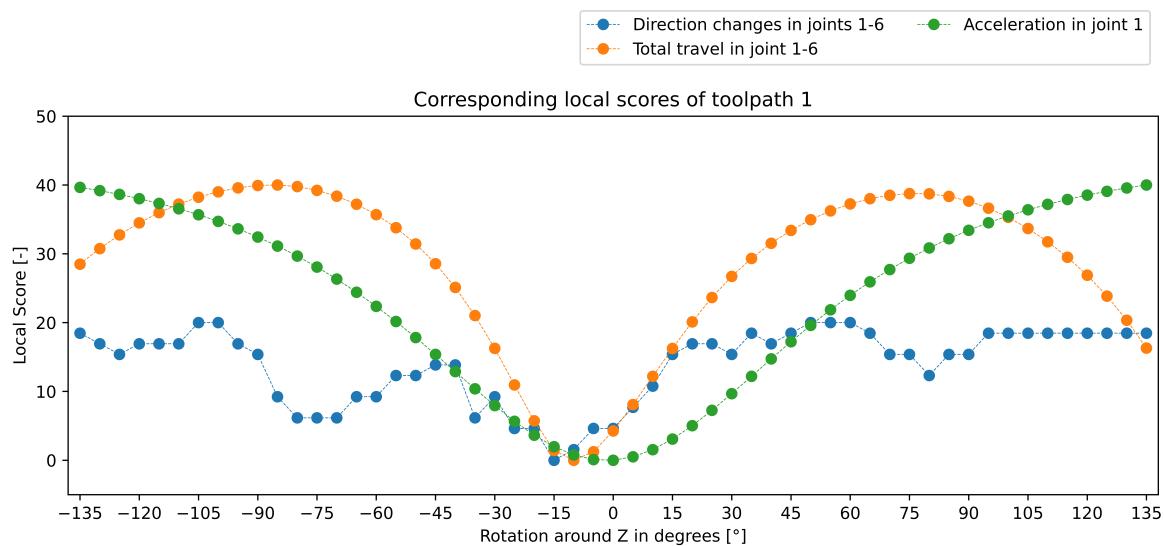
Table 1.2: Selected process variables and their importance factors

Since only one redundant DoF is being analyzed, it is possible to represent the individual local scores and global score as a one-dimensional graph. Firstly, toolpath 1 is analyzed by incrementing the redundant DoF (C-axis) by 5 degrees, starting from -135 degrees and ending at 135 degrees.

Figure 1.10 illustrates a case with a -45° rotation around the Z-axis for toolpath 1. Similarly, Figure 1.11 displays a case with a $+45^\circ$ rotation around the Z-axis for toolpath 1.

**Figure 1.10:** Toolpath 1 with $A=0^\circ$, $B=0^\circ$ and $C=-45^\circ$ **Figure 1.11:** Toolpath 1 with $A=0^\circ$, $B=0^\circ$ and $C=45^\circ$

A total of 55 time-series of joint positions are generated. On average, the inverse kinematics algorithm takes 35 seconds to calculate the joint values for all 3000 coordinates. The process variables are extracted and scaled in relation to each other, as described in Chapter ???. It is important to note that before scaling, the selected process variables are pre-multiplied by -1, as each value should be minimized. The arrays of local ratings are then multiplied by the weights selected in Table 1.2. Subsequently, the local scores of each process variable can be plotted as a one-dimensional graph, as shown in Figure 1.12.

**Figure 1.12:** Local scores of each process variable for toolpath 1

The acceleration in joint 1 and the combined total travel in the joints exhibit a smooth oscillating curve with a decreasing amplitude towards the positive end of the analyzed range. It is worth mentioning that the maximum value that the local score can reach is 40 for acceleration and total travel, while for direction changes, the maximum score is 20. This is due to the assigned importance factors. The direction changes display a less smooth, but still oscillating trajectory.

Next, the local scores are summed up to calculate the global score. The resulting array, displayed in Figure 1.13, represents the global score achieved by varying the rotation around the Z-axis in relation to all other analyzed values. The green cross on the graph indicates the maximum attainable score and its corresponding rotation, compared to all analyzed boundary conditions.

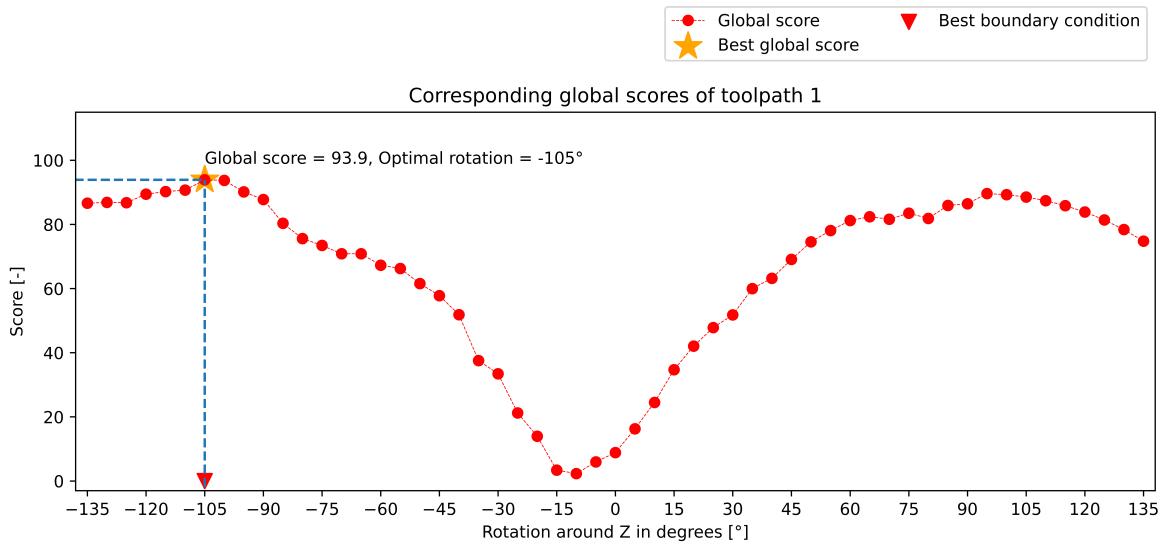


Figure 1.13: Global score for toolpath 1

In this particular case, the highest achievable score is 91.1 at -120 degrees. This nearly perfect score indicates that setting the rotation to -120 degrees results in minimal direction changes, minimal total travel, and close to minimal acceleration in joint 1. It is crucial to emphasize that this rating is only in comparison to the other analyzed boundary conditions.

The same analysis, using identical process variables and weights, can be performed with toolpath 2 and toolpath 3. Figure 1.14 and Figure 1.15 display the global and local scores for each analyzed value of the redundant DoF. For toolpath 2, the best score is 95.3 at -120 degrees, while for toolpath 3, the best score is 89.7 at -90 degrees.

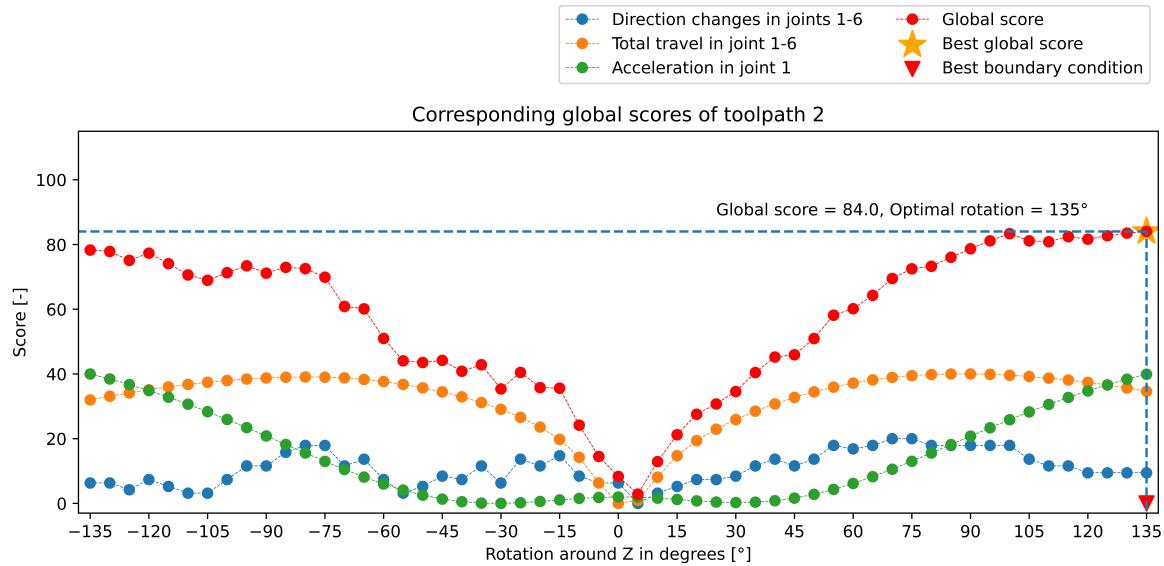


Figure 1.14: Global and local scores in toolpath 2 depending on the rotation around Z

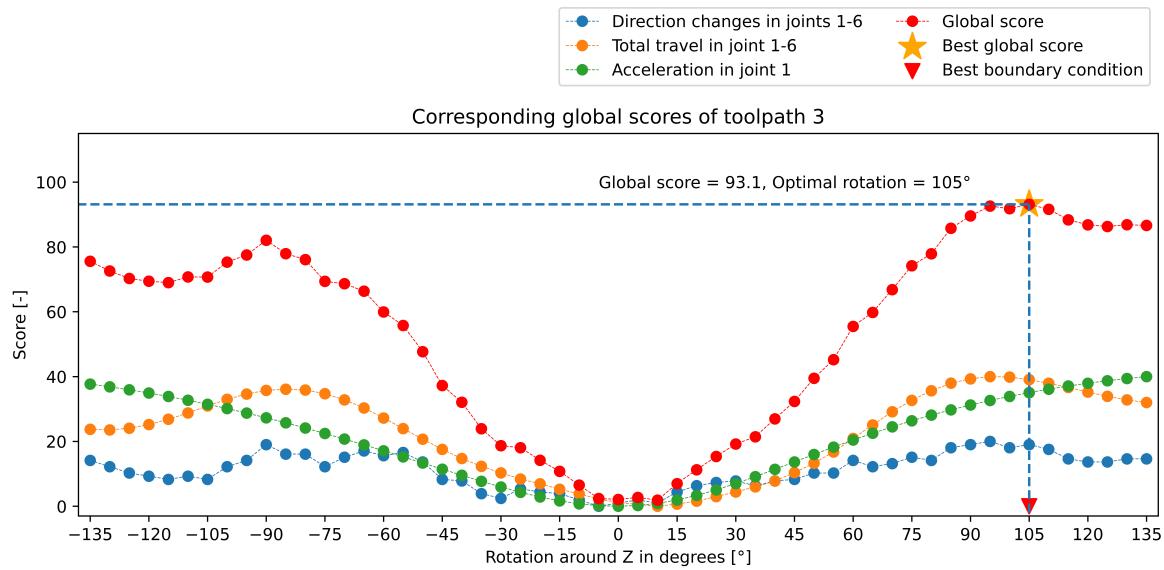


Figure 1.15: Global and local scores in toolpath 3 depending on the rotation around Z

It is noteworthy that even though all toolpaths have significantly different characteristics, all global scores show a similar trend. High global scores are achieved by setting the rotation to less than 0°, while low scores are achieved by setting the rotation to higher than 30°. Additionally, it is important to mention that when a local score reaches its maximum value, it does not imply that the corresponding process variables, such as the number of direction changes, become zero. Instead, it signifies that the number of direction changes is at its lowest compared to all other analyzed options.

Based on this result, it is reasonable to conclude that correctly setting the redundant DoFs can significantly improve the manufacturing process.

1.2.2 Validation on a Production Grade toolpath

In addition to the three simulated toolpaths, a real toolpath used for WAAM is now being used for validation. The toolpath has an organic structure that requires tilting of the rotary-tilt table to ensure that the material deposition process occurs in the direction of gravity. Thus, the position of the rotary-tilt table is not horizontal during the process. The redundant DoF remains the rotation around the Z-axis of the tool, as before. Figure 1.16 provides a visual representation of the analyzed toolpath. It consists of 28,000 coordinates and is modeled to be traversed in 2,800 seconds.

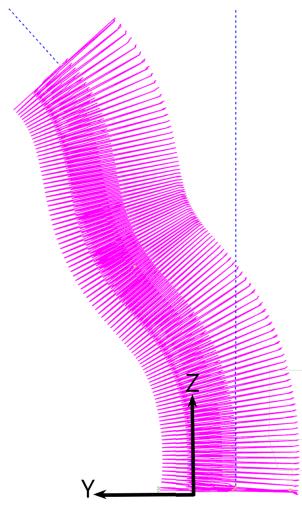


Figure 1.16: Organic toolpath (*Reisch 2023*)

The process variables for the analysis of this toolpath are defined in Table 1.3. In this example, the direction changes in joints 1, 2, and 3 are all individually weighted with a factor of 0.2. The last two variables are the velocity in joint 5 and the total travel in joint 6. These variables are also weighted with an importance factor of 0.2. To obtain a scalar value for the velocity, all elements of the time-series are squared and summed up.

Process variables	Importance factors
Direction changes in joint 1	0.2
Direction changes in joint 2	0.2
Direction changes in joint 3	0.2
Velocity in joint 5	0.2
Total travel in joint 6	0.2

Table 1.3: Selected process variables and their importance factors for the organic toolpath

The redundant DoF is once again analyzed in 5° increments, starting from -135° and ending at 135° . The resulting local scores are shown in Figure 1.17. It is evident that the velocity in joint 5 and the total travel in joint 6 exhibit a smooth oscillating behavior. The direction

changes in joint 1 remain constant for all analyzed settings of the redundant DoF. Therefore, the local score is set to 0 and is not further used in the calculation of the global score. The direction changes in joint 2 and 3 show a slight oscillation and exhibit a non-smooth progression towards the end and beginning of the analyzed range.

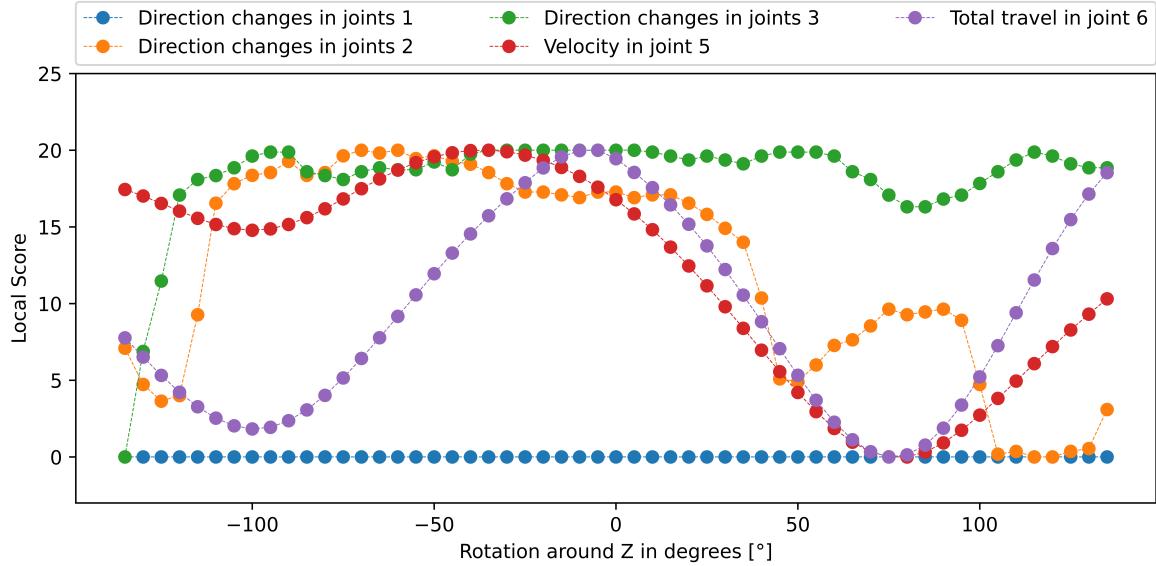


Figure 1.17: Global score for toolpath 1

Figure 1.18 illustrates the combined local scores in the form of the global score. The highest global score is 75.5, which is achieved by setting the C-axis to -15° . Notably, in the range from -70° to 40° , a very smooth curve is observed. This smooth progression of the global score serves as a strong indicator that an optimization algorithm will easily find an optimum setting for the redundant DoF.

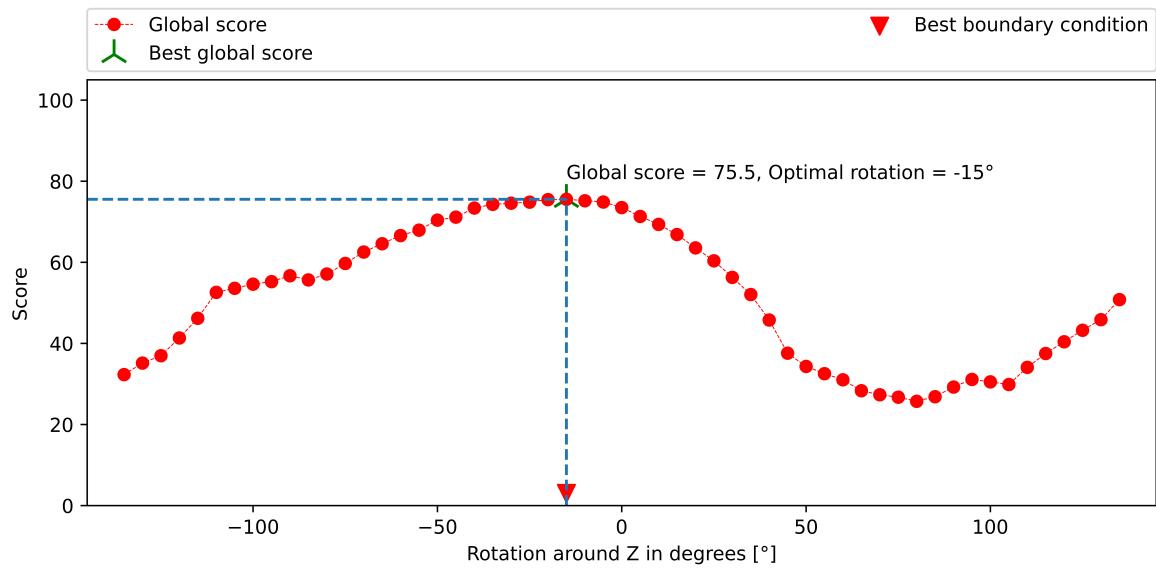


Figure 1.18: Global score for production toolpath

1.2.3 Toolpath Evaluation With two Redundant DoF

To introduce an additional redundant DoF, a rotary-tilt table is simulated. Currently, only the tilting aspect is being analyzed. All coordinates of the toolpath can be rotated by a specified degree around the X-axis of the toolpath coordinate system. Figure 1.19 depicts toolpath 3 with no rotation around the X-axis, while Figure 1.20 illustrates a rotation of +25 degrees.

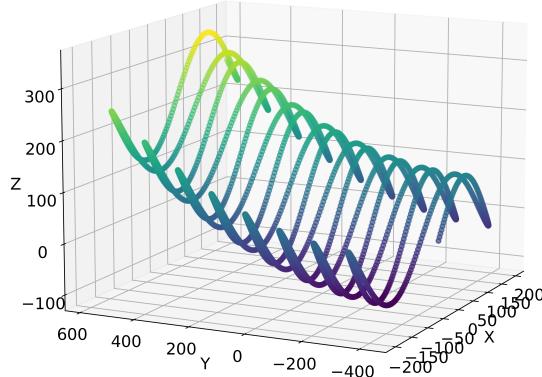


Figure 1.19: Toolpath 3 with no rotation around the X-axis

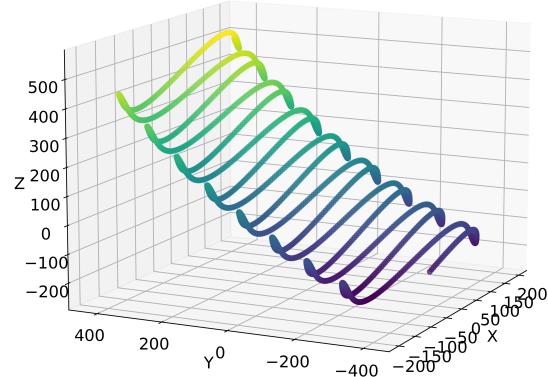


Figure 1.20: Toolpath 3 with a rotation of 25 degrees around the X-axis

Similar to the previous analysis, the same steps need to be followed. The newly selected process variables are presented in Table 1.4. The direction changes of the tilting joints (2+3+5) are combined and treated as one process variable, weighted with a factor of 0.3. Direction changes in joint 1 and acceleration in joint 4 are considered as individual variables, both individually weighted with 0.25. The final variable is the velocity in joint 6, weighted with 0.2.

Process variables	Importance factors
Direction changes in joints 2+3+5	0.3
Direction changes in joints 1	0.25
Acceleration in joint 4	0.25
Velocity in joint 6	0.2

Table 1.4: Selected process variables and their importance factors for 2 redundant DoF

Figure 1.21 displays the robot and its orientation while following the tilted toolpath 3. It is crucial to note that since the toolpath is defined in 5 DoFs in its own frame, the frame of the TCP must also tilt by the same degree as the table. The two redundant DoFs in this case are the rotation around the Z-axis in the frame of the tilted toolpath and the tilting of

the toolpath itself. This introduces an additional dimension, as now two parameters can be adapted for optimization.

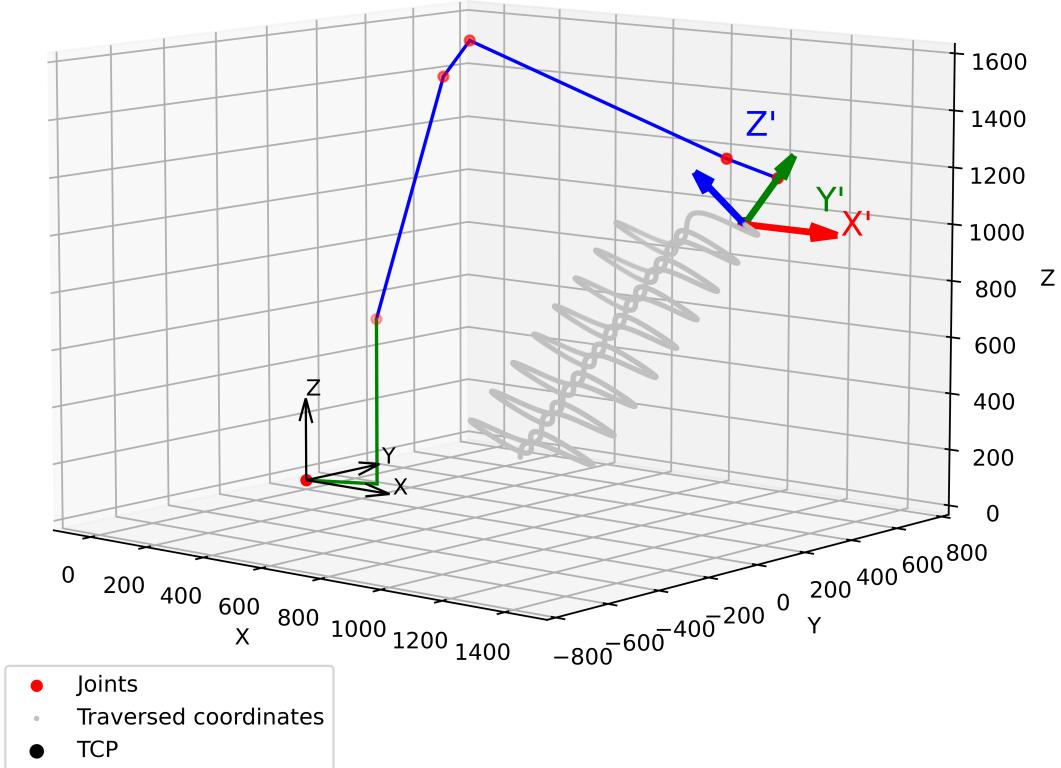


Figure 1.21: Robot following the tilted toolpath 3

The range of possible tilt positions ranges from -45° to 45° in 2-degree increments. For every combination of tilt and rotation, the joint angles are generated using inverse kinematics. To speed up the computation, only every third coordinate is utilized in the inverse kinematic algorithm. This reduces the toolpath by 2000 points and speeds up calculation time. On average, it now takes only 10 seconds to calculate the joint positions. A total of 2530 individual combinations are analyzed.

The extracted process variables are again multiplied by -1, as the objective is to minimize them. Afterwards, the Min-Max scaler is applied. The individual values are aggregated and presented in the form of a matrix. The values of this matrix are visualized in Figure 1.22. The maximum achievable score from all possible combinations is 70.4, visualized by the red cross. This score was attained by setting the table tilt to -5° and the rotation around the Z-axis of the tool to $+35^\circ$. The resulting hyperplane exhibits two distinct local maxima. The entire surface displays a smooth curvature, although it appears less smooth in the range from -20° to 45° .

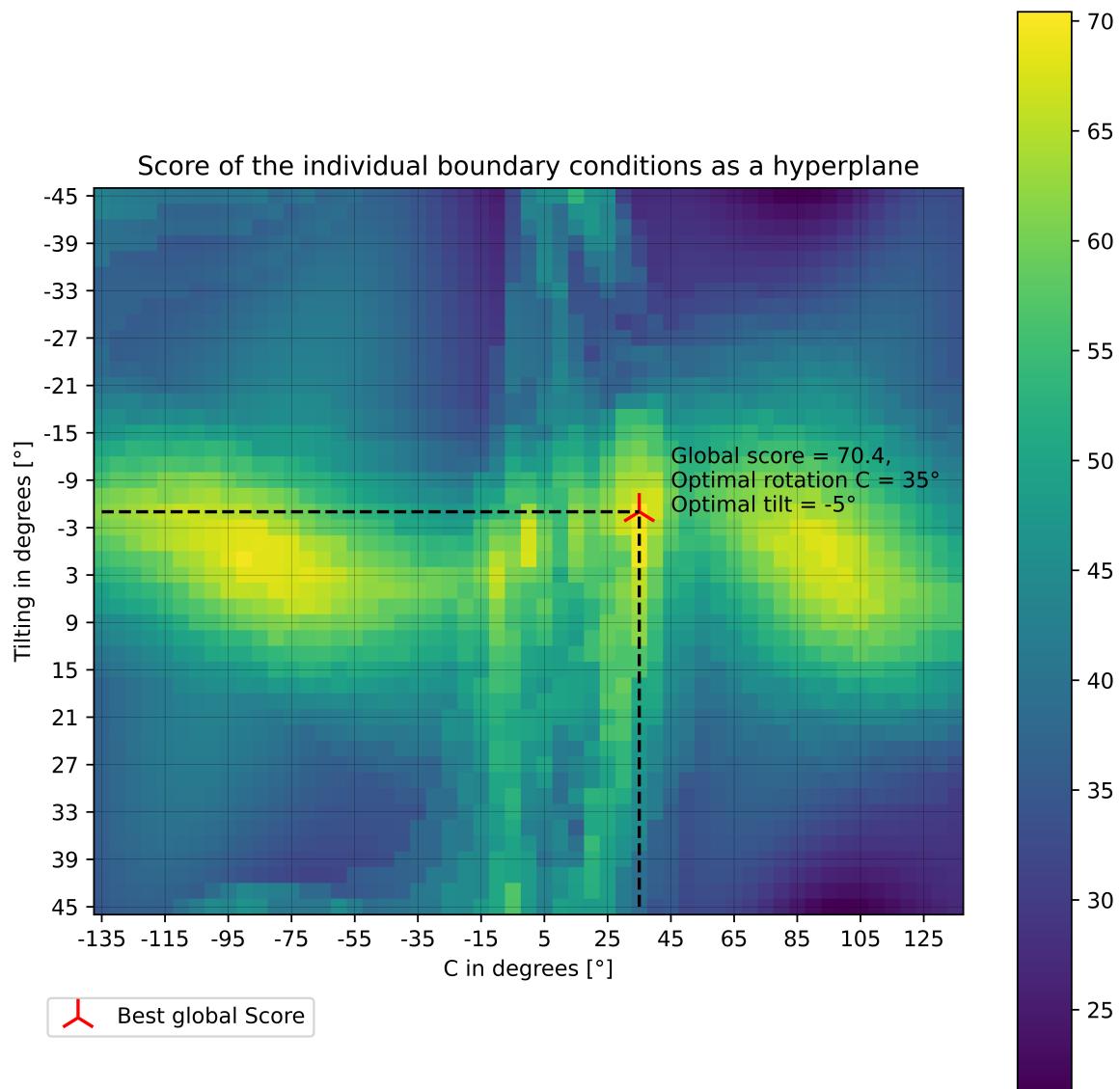


Figure 1.22: Hyperplane representing the global score of toolpath 3

The same analysis is performed with toolpath 1 and toolpath 2. The resulting matrix of the global scores for each possible boundary condition combination is shown in Figure 1.23 and Figure 1.24, respectively. The same process variables and weights (see Table 1.4) are used for the calculations.

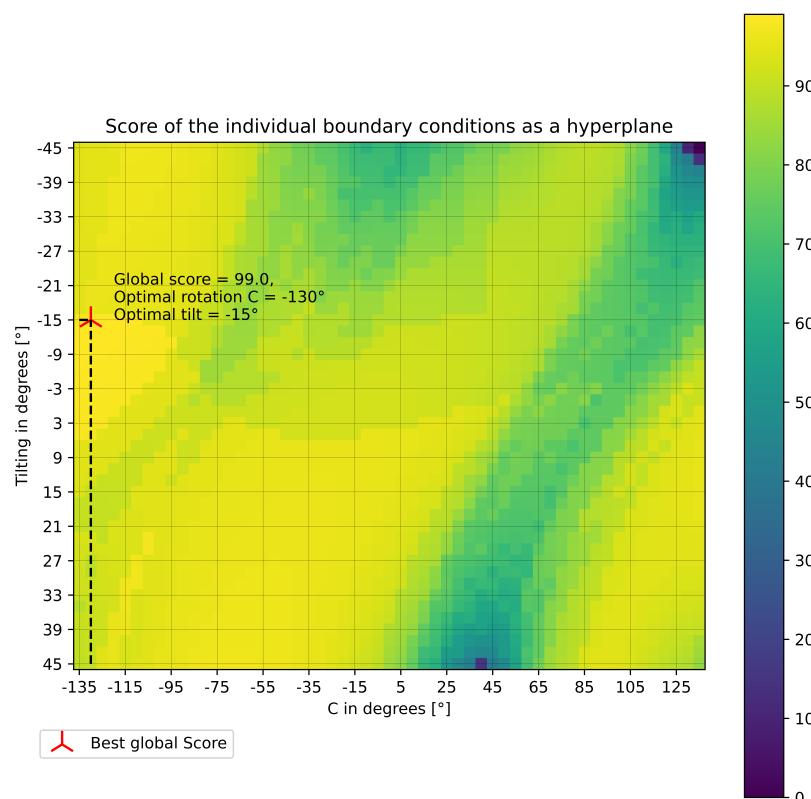


Figure 1.23: Hyperplane representing the global score of toolpath 1

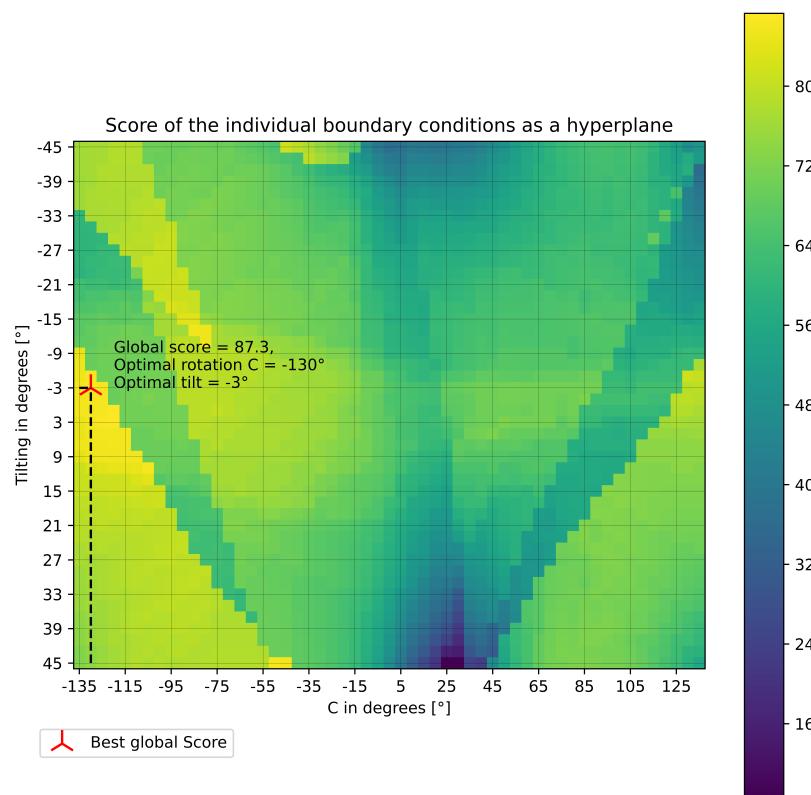


Figure 1.24: Hyperplane representing the global score of toolpath 2

1.2.4 Boundary Condition Optimization

So far, only the analysis of different boundary conditions has been performed by exploring the entire range of possible settings for the redundant DoFs. However, this approach is very time-consuming and becomes exponentially more complex with additional redundant DoF and a finer step size. To address this issue and efficiently search the extensive solution space for optimal values of the redundant DoF, a PSO algorithm is proposed. In this algorithm, individual particles navigate the search space by adjusting their positions based on their own best position and the best position found by the entire swarm. This cooperative behavior enables the particles to explore the search space more effectively and converge towards the best solution (see Chapter ??).

The first test is conducted using the global score matrix of toolpath 3. Initially, 20 particles are randomly placed on the plane. Each particle's score is determined by the corresponding global score at its position. By increasing the number of particles and iterations, the search space can be analyzed more comprehensively. Figure 1.25 illustrates the randomly placed particles on the pre-calculated global score matrix of toolpath 3, considering the previously selected importance factors (see table 1.4).

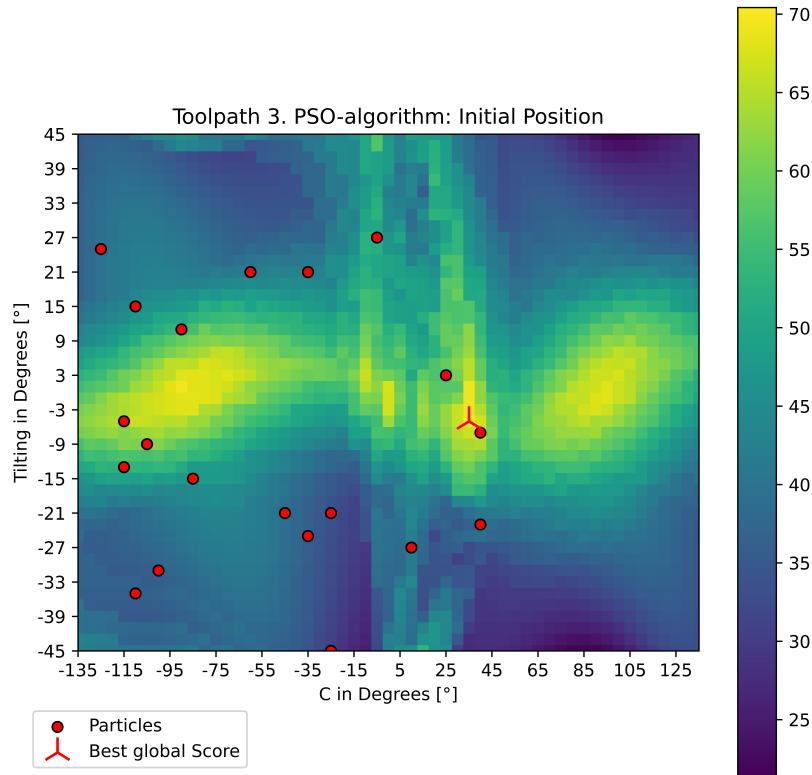


Figure 1.25: Distributed of particles at their initial positions

It is important to clarify that the individual particles in the PSO algorithm do not have access to the entire global score matrix. The visualization of the global score matrix is only provided for the benefit of the human reader to aid in understanding. In reality, each particle only

has knowledge of the global score at its own position. The particles update their positions iteratively by comparing their positions with each other. Each particle determines a new position based on the best score observed so far by all particles and the best score observed by itself.

Figures 1.26 to 1.29 demonstrate the convergence of the particles towards the maximum global score. This convergence is achieved within 5 iterations. It is noteworthy that the best position of all 5 particles corresponds to the global maximum, as discussed in Chapter 1.2.3. This example highlights the ability to explore a high-dimensional space without the need to compute all possible combinations.

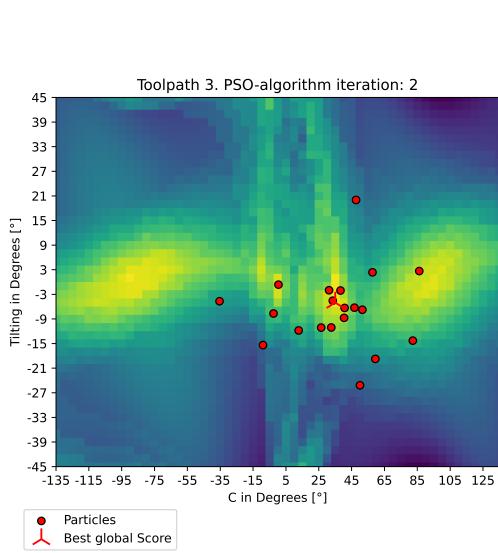


Figure 1.26: PSO Iteration 2 on toolpath 3

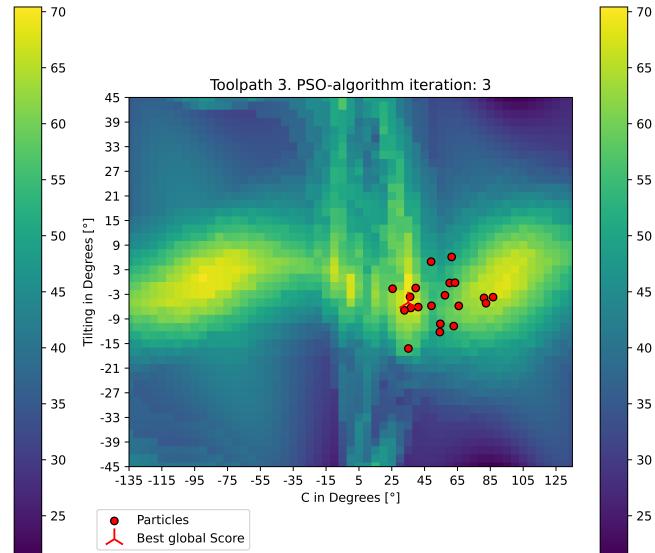


Figure 1.27: PSO Iteration 3 on toolpath 3

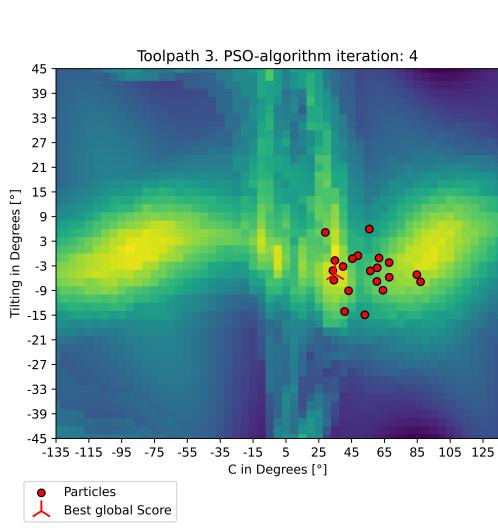


Figure 1.28: PSO Iteration 4 on toolpath 3

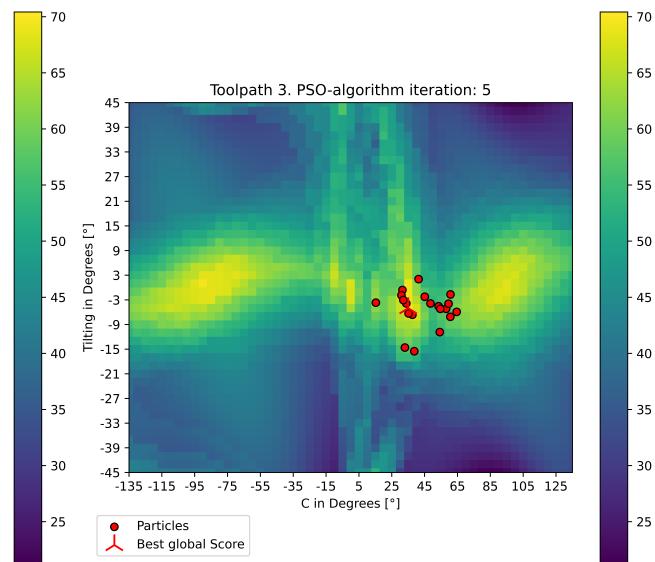


Figure 1.29: PSO Iteration 5 on toolpath 3

One important consideration in this test is, that the scores have already been calculated. The global score matrix is determined by evaluating all possible combinations of different boundary conditions. However, in the intended scenario where this method is used to find the optimal boundary condition, such a pre-calculated matrix does not exist.

Therefore, in each iteration, the scores of the individual particle positions need to be compared relative to each other, taking into account the previous iterations. This process is illustrated in Figure 1.30. Initially, a predetermined number of particles is randomly placed on the plane, with the X and Y values representing the selected boundary conditions. For each selected boundary condition, the joint angles are calculated using the inverse kinematic approach. The analyzed process variables are then extracted and the joint angles are stored. The score for each current position is calculated relative to all stored toolpaths.

It is possible that in an early iteration, a particle had a position with a significantly higher score compared to the other available toolpaths. Therefore, after each iteration, it is necessary to update the score of the particle's most optimal position as more boundary conditions are analyzed. This is done to ensure that a score, which may have been mistakenly chosen as the best, does not influence the subsequent search directions.

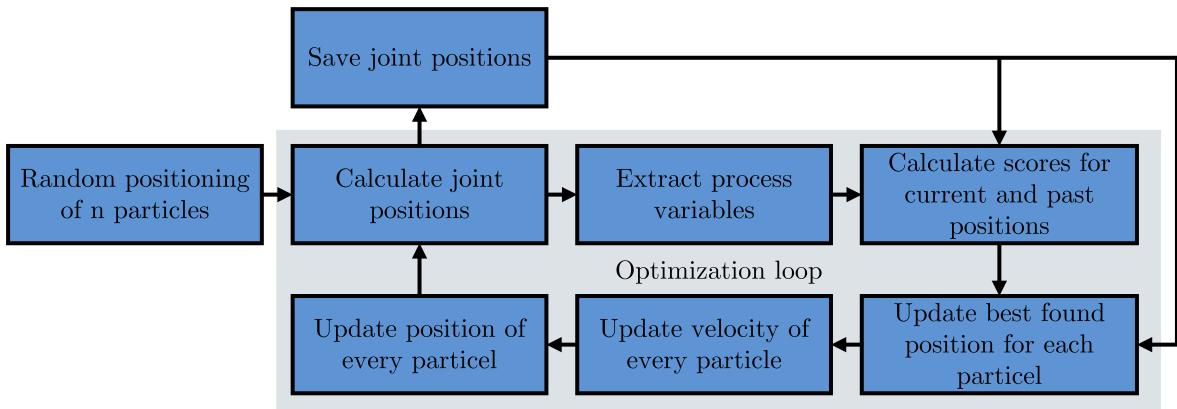


Figure 1.30: PSO-Loop

By utilizing this approach, the following results have been obtained. It is important to emphasize that the precalculated values of the global score matrix are not accessible to the PSO algorithm. They are only used for evaluating the behavior of the particles and aiding in visualization. Figures 1.31 to 1.34 illustrate the evolution of the individual particles. The green circle represents the overall best position discovered by the particles up to that point

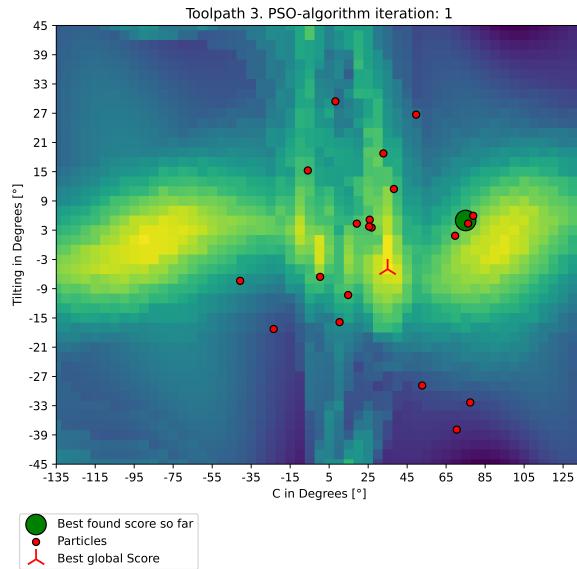


Figure 1.31: PSO Iteration 1 on toolpath 3

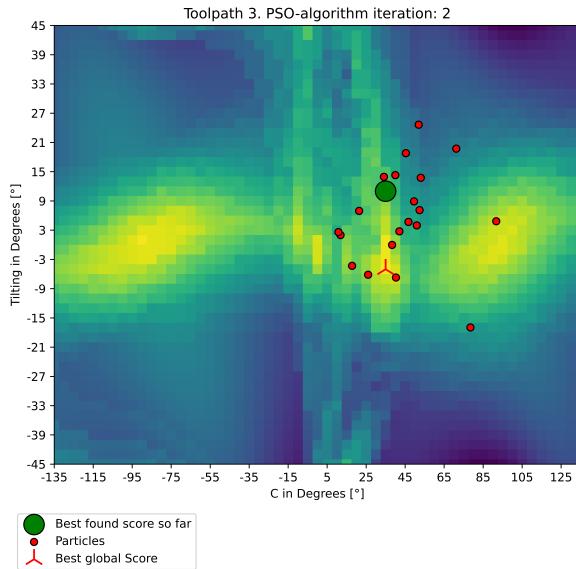


Figure 1.32: PSO Iteration 2 on toolpath 3

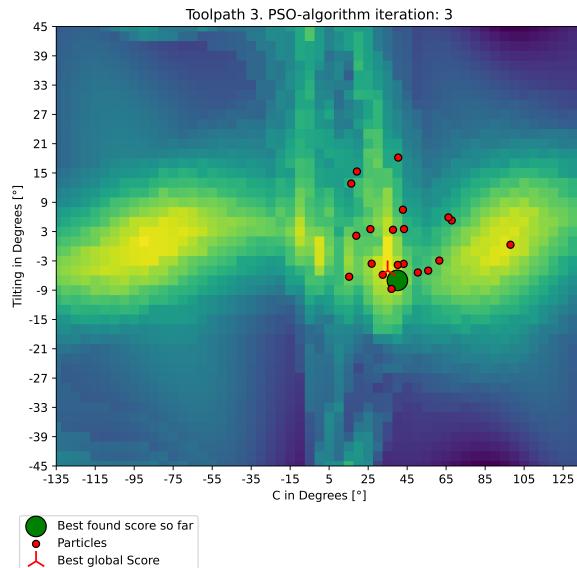


Figure 1.33: PSO Iteration 3 on toolpath 3

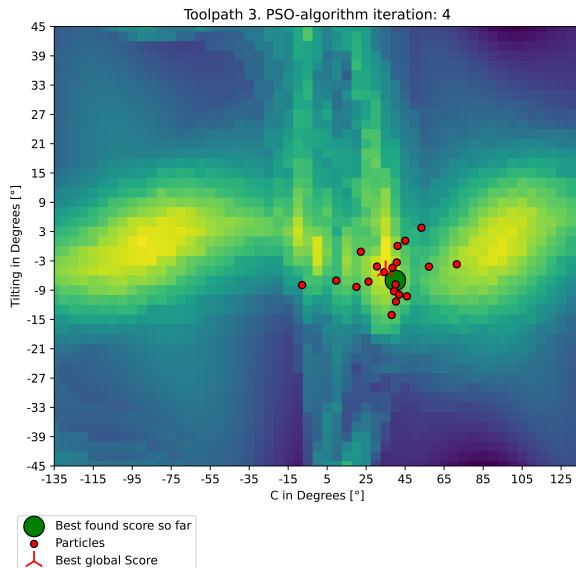


Figure 1.34: PSO Iteration 4 on toolpath 3

The backtracking comparison of all previously visited positions by the particles is clearly evident in the movement of the green dot, which represents the best position found so far. As more points are visited by the particles, the calculation of the best boundary conditions improves. Even without the precalculated global score matrix, the particles are able to converge to the global optimum. Figure 1.35 illustrates the positions of the particles after the fifth and final iteration.

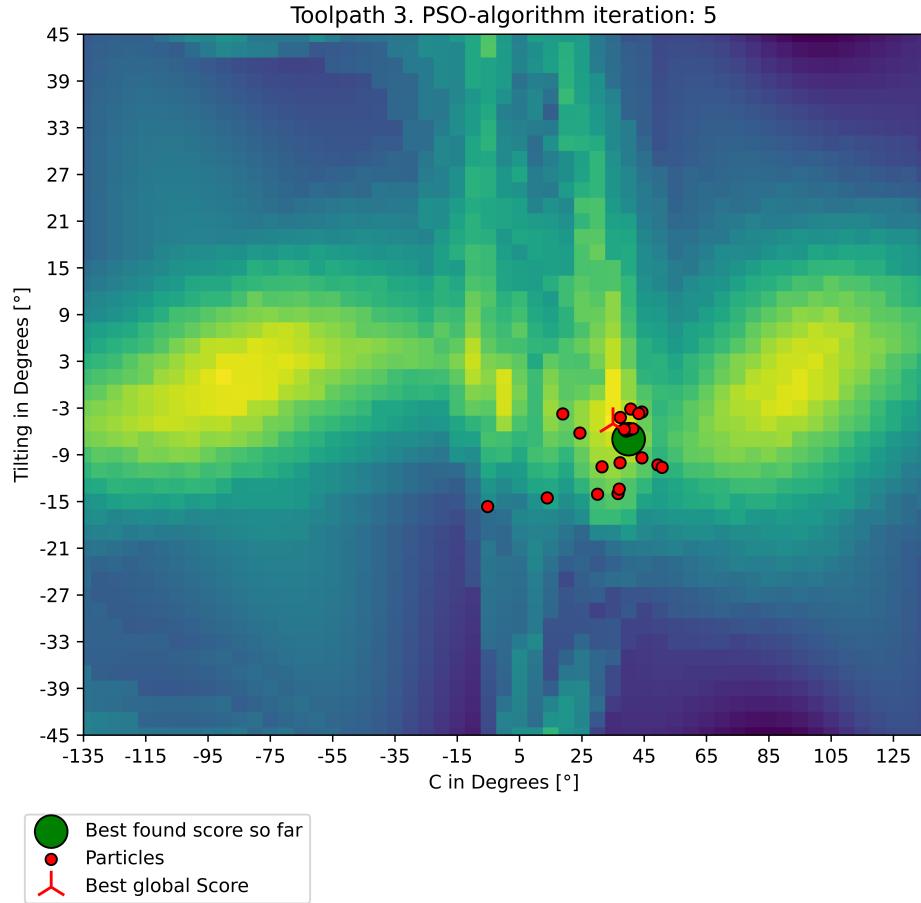
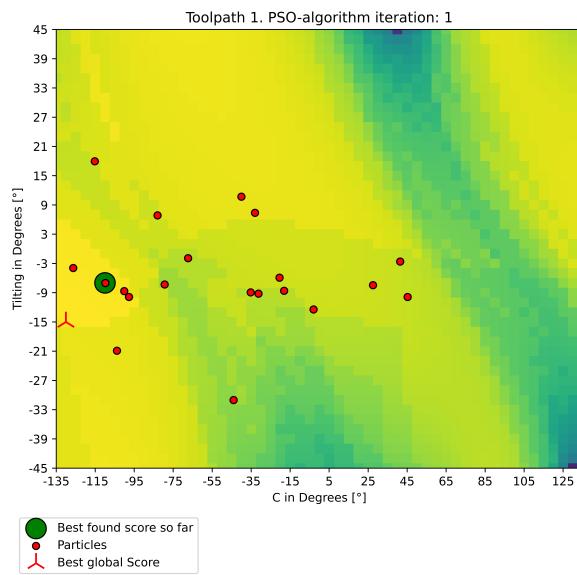
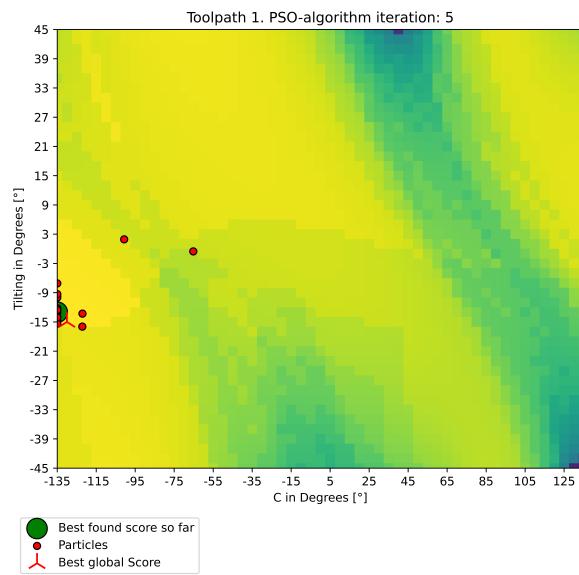
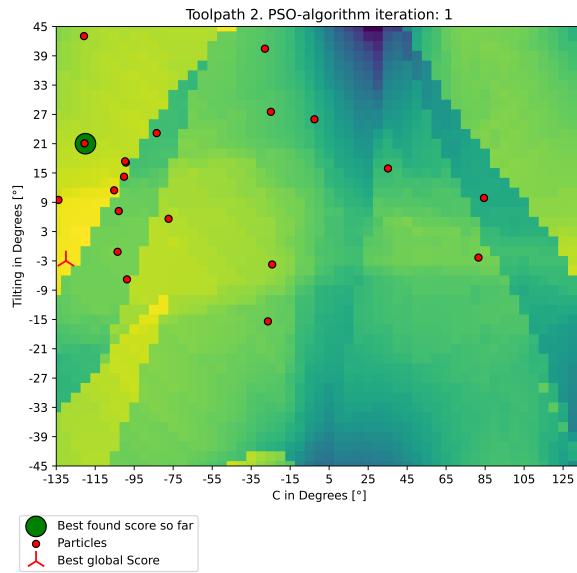
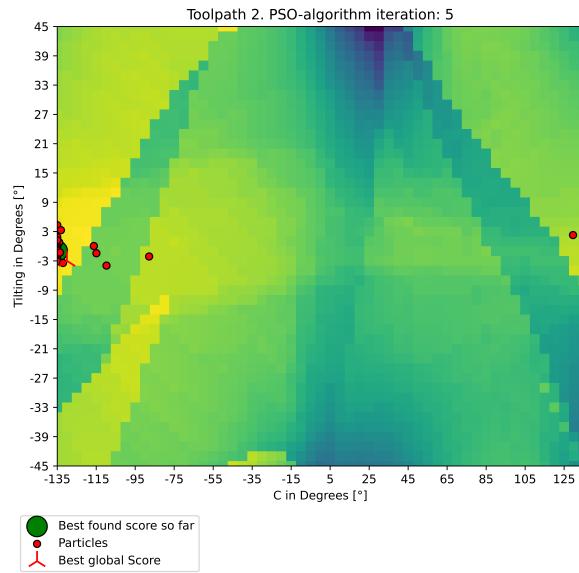


Figure 1.35: PSO iteration 5 on toolpath 3

The same optimization process can be applied to toolpath 1 and toolpath 2. The same process variables and weights are selected as for toolpath 3 (see Table 1.4). Once again, 20 particles are analyzed over 5 iterations. Figures 1.36 through 1.37 depict the positions of the particles after the first and final iteration when analyzing the first toolpath, while Figures 1.38 and 1.39 showcase the initial and final positions of the particles when analyzing the second toolpath. In both cases, the PSO algorithm successfully converges very closely towards the optimal boundary condition.

**Figure 1.36:** PSO Iteration 1 on toolpath 1**Figure 1.37:** PSO Iteration 5 on toolpath 1**Figure 1.38:** PSO Iteration 1 on toolpath 2**Figure 1.39:** PSO Iteration 5 on toolpath 2

1.3 Analysis and Discussion of the Results

In the following, the results are analyzed in detail and critically discussed.

1.3.1 Summary and Analysis of the Results

When analyzing only one redundant DoF, specifically the rotation around the Z-axis (see Figure 1.13, Figure 1.14, and Figure 1.15), it becomes evident that there is significant potential for improvement. In this initial test, the process variables "direction changes in all joints," "total travel in all joints," and "acceleration in joint 1" are analyzed. The local score for "total travel in all joints" and "acceleration in joint 1" both exhibit a very continuous change over the entire range of the analyzed rotation. This characteristic is a clear indication that optimization algorithms are a reasonable choice for finding the optimal boundary condition.

To further validate the proposed method for analyzing process variables, a production-grade G-Code is examined (see Figure 1.17). This G-Code consists of 28,000 coordinates and takes significantly longer to analyze. The resulting comparison of all possible values for the redundant DoF shows a fairly smooth curve of the global score. It is important to note that the local score for the direction changes in all analyzed toolpaths is not as smooth as the other analyzed variables.

When considering the scenario where two DoF can be set, it is observed that toolpath 1 and toolpath 2 have very similar boundary conditions for the global optimum (see Figure 1.23 and Figure 1.24). This suggests that these two toolpaths do not differ significantly from each other when considering the process variables alone.

Analyzing the hyperplane of toolpath 1 (Figure 1.23), it is clearly visible that most of the possible combinations result in a very high global score. The combinations that lead to a low local score form a ravine that is clearly visible on the right-hand side. In the case of toolpath 2 (Figure 1.24), distinct streaks can be observed. The middle of the matrix represents a continuous region, while multiple discontinuities occur on the sides. Toolpath 3 (Figure 1.22) exhibits an almost symmetrical hyperplane. Two local optima are visible on the left and right sides, while the middle part consists of a jagged area with many non-smooth jumps. It is worth noting that the calculation of each individual global score matrix takes approximately 1 hour.

Based on the results obtained from the PSO algorithm (see Figure 1.35), it can be concluded that achieving a close-to-optimal result is feasible when the global score matrix yields smooth surfaces. By implementing this approach, a significant reduction in computation time is possible. Instead of calculating the entire matrix, only 100 toolpaths need to be computed using the inverse kinematics algorithm, resulting in a computation time of just 15 minutes.

The number of particles is selected to be as high as possible while also considering the computational costs. Rather than increasing the number of particles, the number of iterations is

set to 5 in order to facilitate convergence towards the global optima. The results clearly show that such a convergence is possible.

1.3.2 Discussion of the Results

Even though the results show a very promising outcome, it is necessary to consider some additional factors. One of the most obvious elements is that only three artificial and one production-grade toolpaths are analyzed. To validate this method in detail, it is necessary to use more real-life production G-code with correctly modeled robotic systems and analyze whether it can be optimized. It should be noted that this work only provides a limited excerpt and does not analyze complex multi-axis operations, which are a significant advantage and building block of the WAAM process.

The selected inverse kinematics algorithm is not designed for optimal high-performance calculations, making it infeasible to use when dealing with toolpaths that have millions of coordinates. Additionally, the algorithm calculates the joint position numerically rather than analytically, which can result in unexpected robot poses. CAM software such as *Siemens NX* or *Master CAM* offer additional options for inverse kinematics that can be used to fine-tune the behavior of the robot.

When analyzing multiple process variables, it is not guaranteed that the resulting surface will be smooth and optimal for the selected optimization algorithm. Additionally, when working with a PSO algorithm, the final result strongly depends on the initial distribution of the particles. If the optimum is a very tight and sharp spike, the probability of finding the optimal boundary condition is significantly lower. This is particularly true in systems with 3 or more redundant DoF, where simple optimization algorithms can lead to suboptimal results or require unfeasibly long computation times. When implementing the PSO algorithm, 20 particles over 5 iterations are analyzed. In future work, it is necessary to analyze how an increase in population size will improve the convergence rate.

When analyzing Figure 1.13, Figure 1.14, and Figure 1.15, it becomes evident that the best boundary conditions for the highest global score are very similar. Both toolpath 1 and 2 have an optimum of -120°, while toolpath 3 has an optimum at -90°. This suggests that there could be an optimal robot configuration that is applicable to any toolpath.

In general, the presented methodology does provide a solid basis as a proof of concept for the proposed method. However, additional implementations and test are necessary for it to be feasible in an industrial environment.

Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Garching, March 01, 2024

(Signature)