

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

1st Marius Breuer

Siemens

T AMC FMP-DE

Munich, Germany

marius.breuer@siemens.com

2nd Jan Nalivaika

Technical University of Munich

Munich, Germany

nalivaika@outlook.de

3rd Ludwig Siebert

Technical University of Munich

Institute for Machine Tools

and Industrial Management

Munich, Germany

ludwig.siebert@iwb.tum.de

Abstract—This paper discusses a methodology for analyzing and optimizing process variables in manufacturing systems with redundant DoFs. The main focus lies on a 6-DoF industrial robot that traverses a toolpath defined in 5-DoFs (X,Y,Z,A,B). This means that the rotation C, which represents the rotation around the tool symmetry axis, is a redundant DoF and can be chosen freely. The first part of the methodology covers the analysis of process variables, such as direction changes in the joints or the total travel of the joints. The goal is to understand how the redundant DoFs affect the robot's behavior and how they can be used to optimize user-selected process variables. The second part of the methodology proposes a procedure to optimize the user-selected process variables by properly constraining the redundant DoF. After modeling the robot in Python, validation is performed on multiple toolpaths. The results show that the redundant DoFs can significantly affect the process variables. By implementing a PSO algorithm, it is possible to find the ideal setting for the redundant DoFs, thus maximizing the process quality with respect to the process variables.

Index Terms—Industrial Robot, Redundant DoF, Milling, WAAM, Optimization, Process variables

I. INTRODUCTION

In the age of "Industrie 4.0", advanced technologies like digital twins, have greatly transformed industrial manufacturing [1]. A considerable amount of data can be gathered from various processes, like milling or 3D printing. By analyzing this data, it is possible to find new and optimized methods for improving efficiency as well as streamline and enhancing the manufacturing process [2]. Computer-aided manufacturing (CAM) has been introduced as a crucial tool to improve productivity and accuracy in creating customized products [3]. CAM systems automate and optimize tasks such as machining, welding, and assembly [4]. One of the most important areas of CAM is the calculation of the tool path for computer numerical control (CNC) machines as well as the movement and behavior of multi-axis industrial robots [5].

Industrial robots are a dominant part in the area of manufacturing as they can perform multi-axis movements. These capabilities are especially helpful to fulfill the customers wishes for individualized products and one-off parts [6]. They

are cheaper to acquire and easily adapted for different tasks compared to CNC milling machines, but have their own set of disadvantages like for example lower stiffness [7, 8]. One of the most important advantages of industrial robots is their wide adaptability. They allow for quick reconfiguration to produce different components or products, promoting flexibility in manufacturing [9].

Achieving better efficiency and continuous sustainability in the current fast-changing environment, requires a thorough analysis of the interdependent relationships between the manufactured part, the user-defined process parameters, and boundary conditions that govern multi-axis robot programs [5, 10]. Companies that work with industrial robots can place a strong emphasis on energy reduction, cycle-time minimization, or precision. By defining optimal boundary conditions, it is possible to optimize these process variables. CAM enables the simulation of the planned process, thus adapting any boundary conditions in advance, to fit the selected goals [11, 12, 5, 13]. This publication elaborates a approach for analyzing process variables and optimizing boundary conditions in multi-axis robot programs.

II. STATE OF THE ART

A. Problem Formulation

Manufacturing systems that incorporate redundant DoF offer significant advantages in terms of flexibility and adaptability [14]. One example of a system with redundancy is a 6-DoF industrial robot with a rotary tilt table, which brings the system to eight DoFs. However, these systems also present various conflict points that need to be carefully managed to ensure optimal performance [15].

One of the critical challenges in manufacturing systems with redundant DoFs is singularity avoidance [16, 17]. Singularities, which are critical points in the motion of a robot manipulator, arise when the system loses its ability to maintain full control or achieves limited mobility as a consequence of specific joint configurations [18]. These configurations result in the loss of a DoF or make the system highly sensitive to small changes, leading to unstable or even unpredictable and

dangerous behavior [19, 20]. Limiting the possible positions by adding artificial constraints can help to avoid this problem [21]

Another significant aspect of manufacturing systems with redundant DoFs is joint acceleration and jerk, which is the rate of change of acceleration. The robot must allocate accelerations effectively among its joints to achieve smooth and coordinated motion. Failure to do so can result in jerky or erratic movements, which not only compromises precision but also impacts the efficiency of the manufacturing process [22]. Rapid changes in acceleration and jerk can cause mechanical stress and thus, decrease the system lifespan. Additionally, the joints can be limited in their ability to keep up with the required speed due to limitations in power [23]. As of now, no publication is considering the option of constraining the redundant DoFs, to effectively mitigate the occurring joint accelerations while at the same time considering further process variables.

Precision is a crucial element in manufacturing systems, and closely tied to its stiffness. The robot needs to have precise control over the movement of each joint to achieve the desired accuracy in the manufacturing process. However, achieving and maintaining high accuracy and repeatability can be difficult due to the increased complexity [22]. Frequent changes in direction in the joints are another factor that affects precision. Due to the serial kinematics of industrial robots, the present play in the motor joints can add up the inaccuracies and impede the manufacturing process [24, 25]. Mechanical stresses, decreased precision, and increased energy consumption can all result from abrupt and frequent direction changes [26]. Furthermore, effectively coordinating the movement of multiple joints to execute rapid direction changes can prove to be a computationally intensive task [27]. Poor robot configurations can result in prolonged and unnecessary movements, ultimately hampering the overall productivity of the manufacturing process [28]. If constrained effectively, the redundant DoFs can serve as helpful tool to reduce the direction changes in the joints of a industrial robot, while its execution of a toolpath.

While redundant DoFs may introduce potential conflicts and require special attention, they can also significantly enhance performance in manufacturing systems [29]. The added DoFs increase flexibility and adaptability, enabling the robot to carry out complex tasks more efficiently. Redundancy enables multiple approaches to achieve a desired end-effector position or orientation. By effectively utilizing the surplus of DoFs, manufacturing systems can enhance their performance, increase efficiency, and exhibit greater process stability while performing diverse tasks [15].

Currently, there is no integrated system that can evaluate a computed tool path based on the configuration of the manufacturing machine as well as the chosen process variables, such as combined direction changes or stiffness. Additionally, there is no published method that provides an optimal or near-optimal solution for defining the necessary constraints for a specific goal like for example, minimizing energy usage while at the

same time reducing joint accelerations.

B. Comparison of Publications

In the following, a literature analysis is performed regarding the optimization of various process variables. The focus lies on manufacturing systems with redundant DoFs, specifically for tasks such as milling and WAAM. The main focus lies on the process variables: singularity avoidance, joint accelerations and jerk, stiffness and energy use. In cases where no literature is available that incorporates redundant DoFs, non-redundant systems are analyzed. Additional variables like precision and maximum load capacity can also be analyzed but are omitted from the detailed analysis due to the limitations of scope.

1) Singularity Avoidance: As mentioned in Chapter II-A, singularities occur when the robot manipulator loses control or achieves limited mobility due to certain configurations [18]. This results in the loss of a DoF or makes the system highly sensitive to small changes [19, 20].

Due to the infinite possible solutions for the inverse kinematics of redundant manipulators, it is exceptionally challenging to predict and prevent the occurrence of singularity configurations during motion planning [30]. In tasks that involve functional redundancy, as where the manipulator has more DoFs than required for the task, the general projection method cannot be applied [31]. Robotic industrial welding processes often have functional redundancy due to the presence of symmetry axes when using generic 6-DoF industrial robots. Different approaches have been proposed to solve functional redundancy, including adding a virtual joint to the manipulator or using the twist decomposition approach (TWA).

Most of the research is limited to the mathematical analysis of singularities and does not consider the industrial implementation of the proposed algorithms in an industrial setting. The manipulability measure and maximization of Jacobian minors are commonly used methods to avoid singularities. Other methods, such as condition number and singular value decomposition, can also be used [32]. Another mathematical analysis performs a differentiation between non-recoverable singularities and configurations where through self-motion recovery into a non-singular configuration is possible [33].

Another approach proposes a kinetostatic performance index for evaluating the quality of robotic postures, which includes singularity avoidance and joint limit consideration [34]. This method is also transferable to applications like milling. A parameter called "condition number" and "manipulability" are introduced, which are used to calculate the kinetostatic performance index. The presented method can increase the distance from singularities and lower the maximum rotation velocity of the fourth joint. One disadvantage of the proposed method is the manual selection of a parameter. This parameter is responsible for avoiding joint limits and minimizing joint velocities. Manual fine-tuning of that parameter is required for optimal performance.

Further approaches are proposing roll motion around the tool's symmetry axis to counter the loss of a DoF at the singularity. Paths with varying tool roll or fixed roll angles can be chosen, with considerations for tool elevation changes. Selecting paths with a fixed roll angle simplifies implementation for existing robot controllers [20].

Another approach uses the non-square Jacobi matrix and, after analysis, derives a simplified version through the selection of coordinate systems and primary transformation. By using block matrix analysis, the singularity conditions of the articulated robot are determined. A singular configuration avoidance algorithm is used to avoid singular patterns through constraining redundant DoF [30].

Neural networks and other machine learning approaches are commonly used to solve the issue of inverse kinematics. In this case, the optimization variable is not only limited to singularity avoidance but can also be focused on precision or optimization of feed rate [31].

2) Optimization of Joint Accelerations and Joint Jerk:

Jerk and acceleration control are critical because high values can wear out the robot structure and significantly stimulate its resonance frequencies. Vibrations caused by non-smooth trajectories can harm the robot's actuators and produce substantial deviations compared to the desired path. A recently published approach uses an adaptive greedy algorithm for generating the jerk-optimized trajectory with discrete time constraints.

The proposed algorithm improves the trajectory in an iterative manner after obtaining an initial trajectory by utilizing a graph-search method [35]. A further method proposes a sequential quadratic programming method. The results show that optimal time-jerk trajectories with traveling time constraints can be obtained. By modifying the weighting coefficients and the elastic coefficient, it is possible to achieve various trajectory planning objectives, such as rapid execution, a seamless trajectory, or a combination of both aspects [36].

Another method is proposing a method of reconstructing the path by a sequence of via-points that define the positions and orientations of the robot's end-effector. Unlike most minimum-jerk trajectory planning techniques, this algorithm does not force an execution time beforehand and takes into account constraints such as upper bounds on velocity, acceleration, and jerk. The algorithm uses a hybrid objective function that balances execution time and smoothness of the trajectory. The output of the algorithm is a vector of time intervals between consecutive via-points that minimizes the objective function [26].

The last analyzed method is using an algorithm for adjusting the increments of the generalized coordinate vector. By using a pseudo-inverse of the Jacobi matrix and a Taylor's expansion, the robot's acceleration and jerk can be calculated. Results show that when the end effector is closer to the center of the robot, joint jerk increases. It is also shown that if trajectories are designed on the X-Z plane and directed away from the robot's center, the jerk decreases [22].

3) Optimization of Stiffness: Stiffness plays a crucial role in machining with industrial robots. It refers to the ability of a machine or structure to resist deformation under an applied load. In the context of machining, stiffness directly affects the accuracy, precision, and overall performance of the robot. A high level of stiffness ensures that the robot remains stable and rigid during machining operations, minimizing unwanted vibrations, deflections, and inaccuracies [37]. This is particularly important when dealing with high-speed or heavy-duty machining tasks, as any lack of stiffness can result in poor surface finish, dimensional inaccuracies, and reduced tool life.

One method for evaluating the stiffness of a robot is using a newly defined performance index, which is maximized to optimize the robot's posture. The problem is solved using a discretization search algorithm, taking into account joint limits, singularity avoidance, and trajectory smoothness. Each joint of the robot is modeled as a linear torsion spring, which is transferred into a stiffness matrix. This method is applied to a 6-DoF robot that is used for a milling operation. The goal of this method is to set the redundant DoF in such a way that stiffness is maximized. Simulations and experiments on an industrial robot validate the performance index, demonstrating improved machining accuracy using this method [38].

Another approach is working with a dynamic model to reduce the chatter in a milling operation with a 6-DoF robot. By considering the frequency response function, the maximum possible cutting depth, without the occurrence of chatter can be determined. The cutting depth is a function of the redundant DoF. In this case, the redundant DoF is the rotation around the axis of the spindle. An experimental analysis of a full-slot cut is performed. The results show that a significant reduction in chatter can be achieved by setting the redundant DoF to the optimal value [17].

A further publication performs a comparative study of robot pose optimization using static and dynamic stiffness models. The results suggest that the static stiffness model can achieve close to optimal results for pose selection for tasks where the process forces do not approach the resonant frequencies of the robot. It is also discussed that static and dynamic stiffness-based optimizations cannot reduce the deflections of the cutting tool to a range smaller than the robot's repeatability [39].

There are many more methods, like finite element analysis, matrix structure analysis, and virtual joint modeling. To enhance stiffness models, further investigation needs to be conducted. The current state of the art shows a need for standardization in stiffness modeling, as there is currently no universally accepted procedure for establishing such models. Developing a modeling process with standard principles, evaluation indicators, and measuring techniques can simplify the selection and application of modeling methods. Additionally, the application of machine learning techniques, such as artificial neural networks, can be explored for stiffness modeling. Processing experimental data using machine learning algorithms is shown to yield high-precision stiffness models [37].

III. METHODOLOGY

4) *Optimization of Energy Usage:* Energy-efficient usage of industrial robots is essential for achieving cost savings and sustainable manufacturing processes. Manufacturers can achieve this by implementing strategies such as optimizing robot movement paths, reducing idle time, and using energy-efficient components, resulting in significant reductions in energy consumption of their robotic systems. Incorporating advanced algorithms enables robots to adapt to changing conditions and operate at their most efficient levels, optimizing energy usage [13].

One publication by Paryanto et al. analyzes the different methods at different development stages of a production environment in regards to energy optimization. The results show that operating speed and payload strongly influence power consumption [40]. Further analysis in a different publication shows that in a setting where a 6-DoF is used to perform a 5-DoF task, energy savings of up to 20.8% can be expected. The proposed method uses the yaw angle (C-axis) as an optimization variable that can be set to a value in a certain range [15].

Another publication analyzes the general energy consumption of an industrial robot. The results show that cooling and movement speed have the most significant impact on energy consumption. The axis drives are responsible for 23% of the energy consumption. Based on this result, it is shown that optimizing the robot's movement in regards to optimal movement will significantly reduce its energy usage [13].

5) *Summary:* Setting the appropriate boundary conditions in form of constraints, directly impacts the performance and efficiency of a production system. By carefully fine-tuning these values, improved process variables such as singularity avoidance, minimization of joint acceleration and jerk can be achieved. Moreover, optimizing energy usage through the adjustment of variables related to movement speed not only contributes to environmental sustainability but also leads to economic benefits by reducing long-term operational costs. Additionally, the consideration of variables like stiffness and joint limits ensures the safety of both the manufacturing system and its operators.

In conclusion, the careful selection and optimization of process variables play a significant role in achieving optimal performance. The discussed publications, work only on the optimization of individual process variables. A method that takes a holistic approach, is not yet published. Additionally no presented method in those publications gives the user the possibility to weigh the individual process variables in relation to each other and optimize the setting of redundant DoFs towards this user-defined goal. The goal of the following proposed method is to fill this demonstrated research gap.

A. Introduction

The proposed method aims to provide a framework for optimizing various process variables of a manufacturing system in order to achieve a global optimum. These process variables are numerical values that describe the behavior and attributes of a system during the execution of a manufacturing process. One example of a process variable and a manufacturing system, is the combined angular travel of all joints of an industrial robot that is traversing a defined toolpath. The goal of this method is to effectively utilize the redundant DoFs of a system, to optimize the process variables that are important to the user. This method is suitable for robotic milling operations, robotic WAAM processes, and other systems that have redundant DoFs. Implementing this method can improve the overall performance and efficiency, ultimately leading to increased productivity.

The method is divided into two parts. Firstly, for a specific toolpath with user-defined boundary conditions (constraints for the redundant DoF), a set of selected process variables are analyzed. The user has the option to assign a relative importance to the process variables and thus specifying a personal priority rating. Based on the numerical values of the process variables and their relative importance, it can be determined how optimal the set boundary conditions are.

Secondly, the optimization of these boundary conditions is carried out. In this step, the aim is to find the most optimal setting for the redundant DoFs in order to optimize the selected process variables. In other words, the aim is to maximize a score describing the process. This goal can be, for example, the reduction of acceleration peaks in the robot's joints as well as minimizing direction changes of the driving motors.

This method is transferable to any manufacturing system that is used to perform a task with more DoFs than required by the toolpath. In robotic cells specifically, this form of redundancy can be in form of rotary-tilt tables, linear-axes or simply traversing a 5-axis toolpath with a 6-axis industrial robot.

B. General Methodology

In a manufacturing setting with redundant DoFs, multiple elements are in interdependence. These elements include: the toolpath, manufacturing machine, material, part geometry and manually set boundary conditions. It is important to understand how each element is influencing the resulting manufacturing process and what values can be adjusted to achieve an optimal outcome in regards to selected process variables.

The manufacturing machine plays a crucial role in defining the toolpath as it determines parameters such as the total working volume, DoFs, maximum feed rates, and whether the manufacturing process is additive or subtractive. This machine can be a 6-axis CNC machine or an 8-DoF industrial robot. The redundant DoFs, which are not bound by the toolpath, are especially important, as they serve as the future adjustment point for the process variables.

The part geometry refers to the finished geometry as designed in CAD, while the material is the user-defined element from which the part is manufactured. These elements, along with the machine, directly influence the toolpath that is then used in the manufacturing process. For instance, the machine's specifications determine whether the spindle or the work piece itself needs to be tilted during machining to achieve the desired geometric features.

In addition to the machine, part geometry, and material, there are further factors that impact the toolpath. These include the availability of end-mills, the desired depth of cut, the machining strategy, and the operation sequence. These factors are considered as adjacent parameters within the overall manufacturing process. Since the toolpath is a movement relative to the work piece, the user must specify additional parameters before initiating the manufacturing process. One example is the positioning of the raw stock material or substrate plate within the machine and establishing the coordinate system that serves as a reference for the TCP. These settings are also part of the adjacent parameters and need to align with the capabilities of the machine. Correct settings of these parameters may necessitate a comprehensive understanding of both the machine and the specific manufacturing process being performed.

The next element that must be defined, are the constraints for the redundant DoFs. These settings are referred to, as user-defined boundary conditions. A straightforward example to illustrate this constraining is when a 6-DoF robot is used for milling operations. In milling, the TCP position is determined by three coordinates: X, Y, and Z, as well as the tilting and inclination around the X- and Y-axes. Thus only five out of six DoFs are defined by the toolpath. For a full specification, the last DoF which is the rotation around the Z-axis, must be defined manually. With the addition of this constraint, the pose of the robot is fully defined. Since the spindle is rotationally symmetric around its axis, its setting does not affect the toolpath. The rotation around the Z-axis can theoretically be set to any value, but can significantly affect the overall process. In practical applications, this rotation value is limited due to factors such as cable routing to the spindle or joint limits. These same limitations apply in WAAM, where the wire feed system combined with the torch position restricts the possible orientations that can be achieved. Once the constraints are established and the toolpath is generated, various process variables can be analyzed

The goal is now to optimize these process variables by selecting the most optimal boundary condition. Some notable variables include the total angular travel of a specific joint or the occurring angular acceleration. Alongside these resulting numerical values that describe the process, the user has the option to assign specific importance or weight to those variables. By weighting selected process variables, an overall objective with a preference of the user is defined. For example, reducing acceleration peaks is twice as important as reducing the number of direction changes in the joints. With the help of the defined importance factors, a score for the determined

toolpath with the set boundary condition can be calculated.

The flowchart displayed in Figure ?? illustrates the mentioned interdependence of the different elements and the resulting process variables. In the following steps of the method, the adjacent parameters, machine, part geometry and material are excluded in the presentation, as they considered fixed elements and do not directly impact the optimization of the manufacturing process as explained in Chapter ?. These elements serve as hard constraints that define the feasible toolpath and are not directly linked to the redundant DoFs. However, it is important to acknowledge that these adjacent parameters can still greatly enhance areas such as cycle time or surface finish of the manufactured part. The main focus of the presented method is placed on the user-defined boundary conditions, user-defined weights, and the resulting process variables.

C. Process Variables

Table I presents an overview of a select group of the various process variables and their numerical form. These process variables can be derived from a toolpath with defined boundary conditions that is executed by an industrial robot. Due to the limited scope of this thesis, only this selected group of process variables is discussed in detail. Further process variables are mentioned in Chapter ?.

The first group of process variables covers the position, velocity, acceleration, and jerk of the individual joints. These parameters are closely related to the joint position, as they are derived from it. The second group includes the number of direction changes of a joint during operation and the total travel of a joint. The third group comprises the continuous and total energy usage of the manufacturing system during production. The last group consists of the reachability index, singularity analysis, and torch orientation. In the following section, each of these process parameters, their numerical form and required post processing is discussed in detail.

TABLE I
PROCESS VARIABLES AND THEIR NUMERICAL FORM

Process variables	Numerical form
Angular position of each joint	Time-series
Angular velocity of each joint	Time-series
Angular acceleration of each joint	Time-series
Angular jerk of each joint	Time-series
Direction changes of each joint	Scalar value
Total travel of each joint	Scalar value
Continuous energy usage	Time-series
Total energy usage	Scalar value
Reachability index	Binary value / Time-series
Singularity Analysis	Scalar value / Time-series
Torch orientation	Time-series

Position, Velocity, Acceleration and Jerk : One of the primary variables in a robotic manufacturing system is the joint positions. This process variable is recorded as a one-dimensional array comprising the rotational position or extension values of each rotary or linear joint at each time step.

This data serves as the foundation for calculating subsequent variables like velocity, acceleration, and jerk. Analyzing these variables is crucial to ensure that the joints are not excessively strained during the manufacturing process, thus prolonging their service life as much as possible. To extend the lifespan of an industrial robot, it is essential to take into account the load on individual joints.

Direction Changes and Total Travel: An important factor in assessing joint load is the number of direction changes that occur during operation. High-frequency rotational changes can lead to degradation and reduced precision in manufacturing processes. This process variable, referred to as the number of direction changes, is a scalar value that can be derived from the angular position of each joint. By analyzing the joint position data further, the total travel of a joint can be determined by integrating the joint velocity over time. Furthermore, it can be important to analyze if velocity changes consistently occur at the same position, leading to wear on the same tooth flank of the gears. This can cause significant localized wear and shorten the lifespan of the joints. Programs or toolpaths that require less overall joint travel are generally preferred. By minimizing the number of direction changes and optimizing joint travel in regards to acceleration and jerk, stress and wear on the robot joints can be reduced, ultimately extending the overall lifespan of the system.

Energy Usage: Estimating energy usage in industrial robot applications is increasingly important in the current manufacturing environment. One accurate method to estimate energy consumption is through multi-body simulation (MBS), which requires a correct 3D model incorporating information about the weight and distribution of the robot joints. Another approach is to utilize Machine Learning (ML) techniques in combination with the joint positions over time or other intermediate analyses. Moreover, in cases where the industrial robot is used for WAAM, the power required for welding can be determined by analyzing the duration during which the welding torch remains active, which can be extracted from the G-code. Similar to the variables mentioned earlier, the continuous energy usage is also stored in the form of an array to capture variations over time.

Total energy usage, measured in kilowatt-hours (kWh), is a crucial variable that can be directly extracted after finishing the manufacturing process. It offers valuable insights into the overall energy consumption of the industrial robot system. This variable can be obtained by recording the overall energy usage as a single value or by integrating the time-series data of continuous energy consumption. By analyzing energy usage, manufacturers can identify energy-intensive processes or operations, optimize energy consumption, and implement strategies to reduce overall energy consumption. This not only leads to cost savings but also provides environmental benefits.

Reachability: The reachability index is a binary variable used to assess the feasibility of executing a program in an industrial robot system. It determines whether all the necessary points defined in the toolpath are within the robot's working volume and can be reached by its TCP. This variable ensures

that the robot can physically access all required positions in the work area. If any point is found to be outside the reachable workspace, adjustments are necessary. Reachability can also be influenced by additional factors such as cable routing. Even if all the traversed points are within the robot's working volume, factors like wire-feed systems or optical fibers used for laser transmission may have limitations on bending degrees. When analyzing cable routing in relation to orientation, the reachability index can be represented as a time-series that records the deviation of the cable's bending angle from the optimal angle.

Singularity Analysis: The singularity analysis variable can be represented either as a time-series or as a single numerical value. It is derived from the smallest eigenvalue of the Jacobian matrix, which is calculated using the current configuration of the robot. This variable can be stored in an array format, capturing the variations in singularity analysis over time. Alternatively, only the smallest eigenvalue encountered throughout the entire toolpath can be recorded. Analyzing the singularity time-series enables the optimization of non-optimal poses, ensuring that the robot avoids singular configurations that may result in reduced performance or unexpected behavior.

Torch Orientation: Torch orientation is a crucial variable in WAAM, as it tracks the tilt angle of the welding torch during the process. Achieving optimal performance requires that the material deposition always occurs in the direction of gravity. When the welding head is positioned upside down, it represents a worst-case scenario where the welding process takes place against the force of gravity. Significant deviation of the tilt angle from the gravity vector makes it challenging to maintain the stability of the molten metal pool, potentially leading to defects such as sagging. To ensure proper monitoring of torch orientation, a time-series is used to record the deviation of the torch angle from the gravity vector. Analyzing this information helps identify any deviations or that may impact the quality of the deposition process.

D. Local Rating and Global Score

In order to evaluate whether the boundary conditions for a toolpath are optimal or can be improved, it is necessary to have a score or rating value that considers the process variables and their importance. Assigning relative importance to different variables can involve subjective judgments, expert knowledge, and consideration of specific manufacturing constraints. For instance, in some cases, minimizing joint jerk may be the main objective, while in others, energy usage may be more important. To quantify the performance of a toolpath, the user can assign weights or importance factors to each variable based on their specific requirements. These weights reflect the relative significance of each variable in achieving the desired optimum. A weighted sum or scoring method can then be used to evaluate and compare the toolpath with varying constraints based on the aggregated scores of the individual process variables. It's important to note that the subjective weighing of variables can vary between different manufacturing scenarios

and requires continuous evaluation and adjustment based on changing priorities or goals.

The score of a toolpath with its boundary conditions is calculated as shown in table II. Each process variable is assigned a local rating ranging from 0 to 100, where 0 represents the least optimal and 100 represents the best-case solution. This local rating is multiplied by the corresponding importance factor, resulting in a local score. The local scores for all variables are then summed to obtain the overall global score for that specific boundary condition. It is crucial to ensure that the sum of all defined importance values equals 1, so that the toolpath with the most optimal boundary conditions yields a global score of 100.

TABLE II
CALCULATION OF A GLOBAL SCORE FOR A SPECIFIC BOUNDARY
CONDITION

Process variable	Local rating	Importance factor	Local score
Process variable Nr. 1	74	0.5	37
Process variable Nr. 2	34	0.1	3.4
Process variable Nr. 3	65	0.1	6.5
Process variable Nr. 4	22	0.3	6.6
Global score			53.5

E. Local Rating Calculation

Calculating a local rating is not a straight-forward approach. The first problem is, that based on a singular value like "direction changes," it is not possible to determine a local rating as it is not clear if that value is close to optimal or far from it. To address this issue, one possible solution is to analyze the same toolpaths with multiple different boundary conditions or constraints, for example different rotation around the Z-axis, and compare the results.

Figure ?? shows how a local rating can be calculated by means of variation. Each variation of the boundary condition results in a different number of direction changes in joint 1 of an articulated robot. The local score can be determined by essentially applying a Min-Max scaler, which normalizes the values between a minimum and maximum value. The range is set from 0 to 100 as described in Chapter III-D.

It is important to note that increasing the number of variations performed before calculating the local score will enhance the accuracy of this approach. If only a few variations are executed, there is a possibility that only similar outcomes will be found, which significantly affects the result. Furthermore, the user needs to specify whether a high or low scalar value of a process variable is desired. The Min-Max scaler should return the local score accordingly.

F. Information Extraction form Time-Series Data

It is important to note that, as discussed in Chapter III-E, the computation of a local score requires transforming time-series data into a single scalar value. This transformation can be accomplished by directly manipulating the time-series data, such as summing up all values, or by conducting subsequent

analyses. Since each time-series captures distinct physical phenomena, each one necessitates a unique process for converting it into a scalar value.

IV. INFORMATION FROM ANGULAR POSITION

In its isolated form, the rotational or angular position of a joint does not provide much information to enable extensive qualitative analysis. However, when supplemented with temporal information, it becomes a valuable source of information. By considering the angular position in conjunction with the temporal dimension, a more comprehensive understanding of joint behavior can be obtained, allowing for more in depth analysis. Figure ?? visualizes what information needs to be added to enrich the analysis of these process variables that are directly related to the angular position of joints.

The first additional required piece of information is the temporal element, which specifies the exact time when a joint should be in a specific rotational position. This information can be recorded in small equidistant time intervals or only record the positional changes. Recording only the change in position is not ideal because it does not accurately represent the physical aspect of the system. In reality, the position cannot change significantly from one time interval to the next. Additionally, this method does not provide information about the rotational velocity at which the joint should change position. On the other hand, continuous recording in small equidistant time intervals can result in a higher number of recorded values and, consequently, a longer time-series. By deriving the position with time, one can determine the velocity, acceleration, and jerk. With knowledge of the maximum capabilities of the motors, such as maximum velocity or maximum acceleration, it is possible to compare whether the toolpath is feasible for execution. If the limits are exceeded, the behavior of the robot will not be as expected.

Figure ?? illustrates the rotational position of a rotary joint over time in degrees. The left side of the figure illustrates the position recorded with equidistant fine grained time intervals, while the right side displays a time-series where only the destination positions and their corresponding times are recorded. Both series have very similar characteristics but offer a significantly different information density.

A. Total Joint Travel and Direction Changes

Variables that can be analyzed without the need for any additional information include the number of direction changes and the total travel of a joint. The total travel can be determined by subtracting the position of two consecutive recorded points and summing up the absolute values. Furthermore, additional information can be derived by separately summing up the clockwise and counterclockwise rotations. By combining the absolute values of these rotations, the overall travel of the joint can be obtained.

Figure ?? provides a visual representation of the calculation of total travel. The total forward rotation (clockwise) can be determined by summing up the lengths of the green

arrows, while the total backward rotation (counterclockwise) is obtained by summing up the lengths of the orange arrows.

The number of direction changes is a variable that can be determined by analyzing the joint position alone, without the need for temporal information. This can be accomplished by identifying points where the position before and after either decreases or increases. However, it is important to note that this method may not be applicable to points where multiple positions are recorded at the same value consecutively. To address the issue of multiple positions recorded at the same value, a potential solution is to introduce a tracking value that indicates whether the previous change in direction of two consecutive positions was either upward or downward. If the direction of two positions is different from the tracking value, the direction change counter is incremented by 1. However, if the direction is the same as the previous points or if the positions are identical (neutral), the direction change counter and tracking values remain unchanged. Figure ?? provides a visual representation of the points where the direction change counter is incremented.

The counters for direction changes and total travel of each joint can be easily transformed into a local rating, as explained in Chapter III-E. This local rating can then be multiplied by an importance factor to calculate the local score. The direction changes can be summed up over all joints, or they can be specifically grouped based on certain criteria or categories.

Table III provides an example of how a global score can be calculated based solely on the number of direction changes and total travel. The table illustrates how different joints and variables can be weighted and grouped together to calculate the global score. By assigning different importance factors to each variable, the overall global score can reflect the desired optimum. It is important to note that only the local ratings are multiplied by the importance factor, not the actual counted direction changes or total travel values, to form the local scores. This allows the local score to range from 0 to 100. In this example, due to the high local rating and importance factor for the direction changes in joint 1, the overall global score is also very high. This clearly shows the significance of the importance factors in the calculation of the scores.

TABLE III
CALCULATION OF A SCORE REGARDING ONLY DIRECTION CHANGES AND TOTAL TRAVEL

Process variables	Local rating	Importance	Local score
Direction changes in joint 1	95	0.7	66.5
Direction changes in joint 2-6	45	0.1	4.5
Total travel in joint 4	34	0.1	3.4
Total travel in joint 1-3 and 5-6	46	0.1	4.6
Global Score			79

In certain cases, adjusting the boundary conditions may not be sufficient to reduce the number of direction changes, but can offer an improvement in a different area. It is important to note that having the same number of direction changes does not necessarily mean that two time-series are identical. Figure ??

visually demonstrates two time-series with the same number of direction changes but distinct characteristics.

To distinguish between these cases in the local rating, the standard deviation can be utilized. By incorporating the standard deviation, toolpaths that result in frequent and closely spaced direction changes can be identified and are generally considered less desirable. By combining the standard deviation of the timely occurrence with the number of direction changes, they can be integrated into a single variable. This can be achieved by dividing the number of direction changes by the standard deviation. This approach provides an evaluation method that takes into account both the quantity of direction changes and the variability within the time-series data. When there are only a few direction changes but with a high variance, the resulting variable value will be very low. Conversely, when there are numerous direction changes but with a low variance, the variable value will be high.

B. Rotation Limits

Additionally, a straightforward analysis of rotational limits can be conducted, requiring knowledge of four distinct values. The first two values are the upper and lower physical limit that a joint cannot exceed, as surpassing those limit can cause significant damage to the robot. The other two values are potential soft limits put in place to prevent over-rotation of the joint beyond its physical limits. To validate whether any joint positions approach or exceed these limits, a simple comparison of all values can be performed. In cases where it is known that a joint is most stable within a specific range, additional limits can be defined accordingly.

Once a toolpath with defined boundary conditions is established, the joint angles can be analyzed through a simple comparison process. If the joint positions exceed the soft limits or deviate excessively from the desired orientation, a "No-Go" exception is triggered. It is important to note that the analysis of rotation limits does not contribute to the calculation of the global score but serves as a validity assessment to determine if the required movement for the toolpath is physically feasible.

Figure ?? visually illustrates an example of how the analysis includes the hard and soft limits, as well as the desired position range, of a specific joint. If the limits are not exceeded, the toolpath with the set boundary conditions is deemed safe to be executed. At each time step, the amount by which the joint exceeds the desired area can be summed up to form a scalar value, which can then be used to calculate the local score. By doing so, the toolpath with the most optimal boundary conditions will ensure that the joint positions remain within the desired area.

Velocity, Acceleration and Jerk of the Joints

To conduct a specific analysis of the rotational velocity, acceleration, and jerk of the joints, a time derivative needs to be applied. By performing simple comparisons of the time-series values, it becomes possible to determine whether the maximum capabilities of the motor driving the joint are being exceeded.

Figure ?? demonstrates how the velocity aspect can be transformed into a scalar value, which can then be utilized in the calculation of a local rating. Firstly, the joint velocity is obtained by taking the absolute time derivative of the joint position. Subsequently, an analysis is conducted to determine the duration for which the absolute velocity exceeds a certain threshold value. In the given example, the threshold is set at 80%.

Additionally, it is feasible to incorporate multiple thresholds and assign linear or exponential weights to them relative to each other. The combined outcome is then utilized to calculate the local rating. For example, if the 80%-threshold is exceeded for 15 seconds and the 90% threshold is exceeded for 5 seconds, the a scalar value is calculated by multiplying the corresponding time and threshold and summing them up. In this case the result is $80 * 15 + 90 * 5 = 1650$. This number is then used in the variation approach to calculate the local rating.

It is also possible to determine whether a short but significant peak over the threshold values is more desirable than a constant but small overstep. Figure ?? provides a visualization of these two cases.

In the former case, where peaks are more desirable, it is sufficient to sum up the area between the velocity and the set threshold. If avoiding peaks is the top priority, all elements in the velocity time-series can be cubed and summed up. By cubing the values, peaks will be exponentially weighted, making them easier to optimize for.

The simplest approach, however, is to square all values and take the sum. This method eliminates the need to define multiple threshold boundaries. Additionally, high peaks will have a greater impact on the resulting value compared to small, constant values, without overshadowing the final result. If the velocity exceeds the maximum velocity, a "No-Go" exception must be triggered, indicating that the movement is not possible. The same rating principle can be applied to acceleration and jerk. Acceleration is obtained by taking the derivative of velocity, while jerk is obtained by taking the second derivative. Individual limits and thresholds can be set to determine the optimality of the robot's movement. If the maximum acceleration or jerk is exceeded, a "No-Go" error is also triggered.

Table IV presents the calculation of a global score, which incorporates a weighting that prioritizes low acceleration in joint 2 and allows for high velocity in all joints. The acceleration in joint 1 and joints 3 to 6 are considered to be of lesser significance. The jerk is completely omitted from the rating in this example. Due to the close-to-optimal acceleration in joint 2 and its high importance value, the overall score of the toolpath with the defined boundary condition is also very high.

TABLE IV
CALCULATION OF A SCORE REGARDING ONLY VELOCITY, ACCELERATION AND JERK

Process variables	Local rating	Importance factors	Local score
Velocity in joints 1-6	45	0.1	4.5
Accelerations in joint 2	90	0.8	72
Accelerations in joint 1 and 3-6	15	0.1	1.5
Jerk in joints 1-6	4	0	0
Global Score			78

D. Continuous Energy Usage

Tracking energy consumption can be achieved through a straightforward method of monitoring the velocity and acceleration of individual joints. The energy demand is composed of two components: the rotational movement of a joint at a predetermined speed and the joint's acceleration.

To determine the energy consumed by each joint, the time-series data of joint velocity and acceleration can be multiplied by the average energy consumption value, either in $\frac{\text{kWh}}{\text{m/sec}}$ or $\frac{\text{kWh}}{\text{m/sec}^2}$, associated with that particular joint. By summing up these resultant time-series, an energy consumption profile for each joint can be obtained. The newly obtained time-series represent the energy consumption to move from one pose to another. Aggregating the time-series data from all joints into one time-series provides an overall estimation of the robot's energy consumption.

This approach offers a notable advantage in its simplicity. Nevertheless, its primary limitation stems from the potential inaccuracies that may arise when working with an average scaling value. If this value is derived by averaging all possible positions, but only a limited number of positions are actually traversed by the robot, significant discrepancies can occur. To address these limitations, more advanced approaches are necessary. For instance, the utilization of multi-body simulations (MBS) in CAM software enables a direct analysis of the exact energy requirements for a robot to move between different poses. This method requires precise modeling of weight distribution to achieve accurate outcomes. However, it is important to consider that implementing this approach may require substantial computation and development time.

Another viable option to consider, is the utilization of a ML approach for estimating energy consumption during transitions between discrete poses. By employing a supervised learning technique, where the input data includes the current joint positions and velocities as well as the target pose, an ML model can be trained to predict the energy required for each transition. This approach offers the advantage of leveraging ML algorithms to provide accurate energy consumption estimates in a more efficient manner. However, it is important to note that generating high-quality training data and training the ML model can be a time-intensive processes.

Figure ?? illustrates the three aforementioned options along with their key requirements. It is important to note that these examples only give an excerpt on how to get an energy estimation and do not cover all possible solutions to address this problem.

Once the time-series data for energy consumption is obtained, it becomes possible to identify peaks and associate them with specific movements. In line with the optimization objective, it is also feasible to establish threshold values, as discussed in Chapter IV-C, to optimize for a constant energy consumption.

E. Total Energy Usage

When the aim is to obtain a single scalar value for the total energy consumption, the same procedures outlined in the preceding chapter can be applied, and the values from the time-series can be aggregated through summation.

In situations where the temporal information in energy consumption is considered irrelevant, alternative ML approaches can be employed. For example, Recurrent neural-networks (RNNs) can be utilized, as they have the ability to take an entire time-series as input and generate a scalar value representing the total energy consumption. RNNs excel at capturing dependencies and patterns in sequential data. By training an RNN model using a time-series dataset, it can learn to predict the total energy consumption based on the provided input. However, it is important to note that, like most ML approaches, this method requires substantial effort and time for generating training data and conducting the training process.

In WAAM, it is important to consider the energy consumption associated with the welding process itself. The determination of energy use in welding involves various factors. The G-code, which comprises instructions for the welding process, can provide information on the wire-feed, voltage and the desired power input. By analyzing this code, it becomes possible to estimate the energy consumption during each welding operation. Alternatively, these parameters are often pre-defined as constants on the welding appliance, with the G-code solely specifying the turn-on and turn-off points. Figure ?? gives an example on how the turn-on (N10) and turn-off points (N70) can be defined in the G-code.

By considering these factors, it is possible to accurately assess the energy usage during welding operations. However, this specific part of the total energy consumption cannot be optimized by defining the boundary conditions of redundant DoFs. The energy required for the welding process is solely defined by the welding parameters and the final part geometry.

V. REACH, SINGULARITIES AND TORCH ORIENTATION

The following discusses the robot poses with respect to reach and alignment, singularity avoidance, and torch orientation. These factors are essential in ensuring successful and efficient robotic operations, especially in WAAM.

A. Reach and Alignment

As mentioned in Chapter III-C, the analysis of the reachability index can be conducted in various formats. The first format involves a simple analysis to determine if all the points that the robot needs to traverse, lie within its work volume without any self-collisions or exceeding the soft and hard joint limits. This aspect is closely related to the joint limits, as discussed

in Chapter IV-B. Additionally, it is necessary to analyze that there are no collisions between the robot and the workpiece. If all these conditions are met, a binary index can be used to indicate the feasibility and safety of executing the program. However, this index cannot be used for optimizing the robot's movement, as the parameters influencing this index are mostly defined by the G-Code.

When utilizing a robotic system with a specific tool, such as a milling spindle or a welding torch for WAAM, it is crucial to consider the boundary conditions and limits associated with that tool. Figure ?? illustrates a case where the rotation around the Z-axis of the welding torch can be manually defined. Each position leads to different strains on the wire-feed system and power-cables.

In most cases, the spindle and welding torch come with cables that provide power from an external power supply. These cables have an optimal orientation where bending and wear are minimized. By positioning the cables in an optimal orientation, the robotic system can operate efficiently and effectively without any interference or limitations due to cable movement, potential damage, or excessive bending or twisting. To assess the optimality of the robot's cable pose, it is important to consider additional information. In certain cases, it is preferable to route the cables in a specific alignment along a spatial vector, allowing for parallel translation or movement in the direction of that vector. In this case, the angle between the planes of the robot's base coordinate system and the optimal translation vector remains constant. Figure ??a provides a visual representation of this example. The wire, shown in red, has an optimal alignment along a vector where the least amount of strain is present. Any position along that vector where the cable transitions to the welding torch tangentially is considered optimal. It is worth noting that this vector can be translated parallel in space while still maintaining its optimal status.

In other scenarios, it is more optimal to route the cables towards a specific point in space, such as a mounting point on a wall or ceiling. Figure ??b illustrates an example where the optimal alignment of a milling spindle is directed towards a designated point where the cables originate. In this case, any position is considered optimal as long as the extended spindle vector aligns with the vanishing point.

To minimize the deviation from the optimal vector or vanishing point, the redundant DoFs can be utilized. One approach is to analyze the deviation and track it over time, creating a time-series. Each element in this time-series can be squared or cubed, depending on the weight assigned to small and constant deviations versus large but short offsets. The time-series is then summed up to form a scalar value, which can be used to calculate the local score through the variation method. Table V illustrates an example of how the scalar value is calculated. It is evident that when cubing the time-series, more importance is placed on the larger deviations rather than small deviations. The deviation in time-step 6 is making up approximately 80% of the summed value when squared and 94% when cubed.

TABLE V
FROM TIME-SERIES OF THE DEVIATION VECTOR TO SCALAR VALUE

Time-step:	1	2	3	4	5	6	7	8	
Deviation:	1°	2°	2°	2°	3°	10°	0°	2°	
Squared:	1	4	4	4	9	100	0	4	Sum: 126
Cubed:	1	8	8	8	27	1000	0	8	Sum: 1060

B. Singularities

In Chapter ??, the concept of singularities in robotic systems is discussed. Singularities occur when the robot's joints align in a way that limits its motion by reducing one or more DoF. To avoid singularities, it is important to optimize the boundary conditions in the redundant DoF. The singularity analysis, as described in Table I, can be represented either as a scalar value or as a time-series that classifies each position based on its proximity to a singularity.

The scalar value can represent the overall smallest eigenvalue of every Jacobi matrix. For this, every pose needs to be analyzed. After calculating the Jacobi matrix, all eigenvalues encountered are examined, and only the smallest one is recorded and used to calculate the local score. This approach allows for an analysis of the overall toolpath rather than specific poses.

Alternatively, a time-series can be used where the determinant, not the eigenvalues, is recorded and stored. By analyzing this time-series, it becomes possible to directly associate the robot's movements with its proximity to a singularity. This enables a more precise understanding of where the robot's motions are close to a singularity, allowing for subsequent optimizations. To transform the time-series into a scalar value, the same threshold method described in Chapter IV-C can be applied.

C. Torch Orientation in WAAM

The torch orientation variable in WAAM, similar to the cable routing orientation variable, analyzes the angle at which the welding torch is positioned during the process. This variable is specific to WAAM and ensures that the torch is at the optimal angle for welding. Achieving material deposition in the direction of gravity is crucial for obtaining the best results. The orientation of the TCP, which represents the welding torch tip, can be determined either through the forward kinematics approach or by extracting it directly from the G-code.

In both the forward kinematics approach and the G-code, the rotation is described as the rotation of the A-, B-, and C-axes. However, the necessary information for torch orientation is simply the angle between the tilted Z-axis of the tool and the vector of gravity. To obtain this information, a dot product is performed between the rotation matrix and the Z-axis of the base coordinate system. This yields a vector corresponding to the tilted Z-axis of the tool. The enclosed angle can then be calculated using the scalar product. This step assumes that the defined base coordinate system is oriented in a way that aligns the Z-axis parallel to the gravity vector. Figure ?? gives a visual example in 2D on how the Z-axis of the tool is deviating from the vector of gravity.

To quantify the torch orientation variable, it is recorded in a time-series format. To obtain a scalar value for this process variable, all the values can be squared, cubed, or subjected to other mathematical operations, and then summed up. This process is similar to the one described in Chapter V-A for the cable routing orientation variable. It is important to note that the tool orientation variable can be in direct conflict with the alignment variable. This conflict is particularly noticeable in cases where the optimal alignment vector does not correspond with the optimal tool orientation vector. In such cases, it is crucial to have knowledge of the parameters of the manufacturing system and adjust the importance factors of the process variables accordingly.

VI. SUMMARY FOR BOUNDARY CONDITION EVALUATION

Figure ?? gives a summary and visual representation, in form of a flowchart, on how a toolpath with defined boundary conditions is evaluated by calculating the individual local scores for the process variables that can be influenced by the redundant DoFs.

It is important to note that for a local score calculation the user defined importance values need to be multiplied with the local rating that results form the numerical values of the process variables. The local scores are then summed up to create the global score.

Figure ?? illustrates how a boundary condition for a toolpath can be evaluated with the mentioned variation method (see Chapter III-E). In this case, the toolpath is defined using only 5 DoFs, including X-, Y- and Z-coordinates as well as the rotation around the X- and Y-Axis. However, the C-axis (rotation around the Z-Axis), which is the axis of symmetry for the tool, is not defined and must be specified manually. Once this boundary condition is established, the joint positions are analyzed and individual variables, such as direction changes, can be extracted. In this example, the objective is to assess the optimality of a rotation angle of zero degrees of the C-axis. To calculate a local score, additional variations of the rotation around the C-Axis are analyzed. With the help of user-defined weights, the local ratings are weighted and summed up to form the global score for the boundary condition (see Chapter ??).

VII. GENERAL METHODOLOGY FOR PROCESS OPTIMIZATION

So far only the analysis of a toolpath with set boundary conditions is discussed. In the following sections, two methods for optimizing boundary conditions towards a specific goal is presented. The main difference between these two methods lies in the different incorporation of a CAM software.

It is important to note that optimizing certain process variables may have a direct negative impact on others. For example, optimizing the tools orientation towards a specific vanishing point can significantly increase the total number of direction changes in the joints. Therefore, the user must be aware of these cross-influences and adjust the weights accordingly.

A. Optimization Without CAM Software in the Loop

Figure ?? illustrates the process of optimizing the redundant DoFs based on a predefined goal determined by weighing the process variables. In this method, the part and the manufacturing machine are fixed components that are loaded into the CAM software. Before generating a toolpath, the redundant DoFs must be set manually. These constraints can be set based on prior experience, as long as they do not lead to any "No-Go" exceptions caused by collisions or exceeding the joint limits.

When the toolpath is generated, it is defined in all available DoFs of the manufacturing machine. This information is now in form of a G-code. To evaluate the set boundary conditions, the information from the G-code is extracted and the redundant

DoFs are varied, resulting in multiple toolpaths with multiple boundary conditions. In some cases, this variation simply involves varying the rotation around the Z-Axis, while keeping all the X-, Y- and Z-coordinates defined in the G-code, unchanged. In other cases, the redundant DoFs involve the rotation and tilt of a rotary-tilt table. In such situations, every single coordinate in the G-code needs to be rotated around the tilting-axes of the rotary-tilt table. After obtaining the multiple toolpaths, each one is subjected to an inverse kinematics algorithm. This algorithm utilizes the machine's parameters, for example, arm length and joint sequence, to calculate the joint positions to reach every point in the different G-Codes. From these values, the score of the originally set boundary condition can be determined. By utilizing a external inverse kinematics approach, the CAM software is not part of the optimization loop. This approach give the user to work with any CAM software.

Following the score calculation, an optimization algorithm (see Chapter ??) is employed to generate a new boundary condition for the redundant DoFs. This cycle of optimization continues either for a predetermined number of iterations or a specified duration.

Alternatively, a desired minimal score can be defined as the target, that the toolpath needs to achieve. Once the most optimal or close-to-optimal boundary condition is found, it is used as input for the CAM software to validate that no collisions or other exceptions are triggered. After this validation process, the G-code can be utilized in production.

In this method, the optimization of boundary conditions occurs externally to the CAM software. The CAM software is solely utilized for generating the initial toolpath and validating the most optimal boundary condition that is returned by the optimization algorithm. The inverse kinematics process is performed outside of the CAM software and can be implemented in various programming languages such as Python or C++. The same is applicable to the optimization algorithm.

The advantage of this approach is that the external algorithms are independent and can be easily exchanged or modified. It does not necessitate in-depth knowledge or access to the source code of the CAM software. Therefore, any CAM software can be utilized with this approach, providing flexibility and compatibility.

However, it is important to note that one drawback of this approach is that the complete closure of the loop can only be achieved if the most optimal boundary condition can be automatically fed back to the CAM software for the final validation. If there is no application programming interface (API) available or accessible for the CAM software, this step may need to be performed manually. This manual intervention can introduce additional time and effort into the optimization process.

Depending on the chosen optimization algorithm, it may be necessary to calculate multiple scores for multiple boundary conditions before generating a new suggestion. In this scenario, the variations in the redundant DoFs can be utilized to calculate multiple scores relative to each other. These scores

can then be used as input for the optimization algorithm, allowing for a more comprehensive evaluation and selection of the most optimal boundary condition. By considering multiple scores, the algorithm can effectively explore different possibilities and make informed suggestions for improving the boundary conditions.

Another drawback of this approach, is that for the variation of the redundant DoFs the G-Code file is utilized. For that a dedicated software-program is necessary that can read the G-Code and adapt the individual coordinates accordingly. Another level of complexity can occur, if the G-Code is utilizing non-standard or user-defined commands. Many edge cases need to be considered before such a program is deemed save for production.

Figure ?? illustrates the process of varying a simple toolpath. In this specific example, the toolpath is defined in six DoFs, with the redundant DoF being the rotation around the Z-axis. Initially, the toolpath is set at its original configuration with $C = 0.0^\circ$. To explore different possibilities, three variations of the toolpath are created by incrementing the rotation by five degrees for each variation. These variations allow for an examination of the impact of different orientations on the overall process. By systematically adjusting the rotation, the algorithm can evaluate the performance of each variation and suggest even better values.

REFERENCES

- [1] Maulshree Singh et al. "Digital Twin: Origin to Future". In: *Applied System Innovation* 4.2 (2021), p. 36. ISSN: 2571-5577. DOI: 10.3390/asi4020036. URL: <https://www.mdpi.com/2571-5577/4/2/36>.
- [2] Morteza Ghobakhloo. "Industry 4.0, digitization, and opportunities for sustainability". In: *Journal of Cleaner Production* 252 (2020), p. 119869. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2019.119869. URL: <https://www.sciencedirect.com/science/article/pii/S0959652619347390>.
- [3] Thomas Feldhausen et al. "Review of Computer-Aided Manufacturing (CAM) strategies for hybrid directed energy deposition". In: *Additive Manufacturing* 56 (2022), p. 102900. ISSN: 2214-8604. DOI: 10.1016/j.addma.2022.102900.
- [4] K. Lalit Narayan, K. Mallikarjuna Rao, and M. M. M. Sarcar. *Computer aided design and manufacturing*. Second printing. Delhi: PHI Learning Private Limited, 2013. ISBN: 9788120333420.
- [5] Zengxi Pan et al. "Recent progress on programming methods for industrial robots". In: *Robotics and Computer-Integrated Manufacturing* 28.2 (2012), pp. 87–94. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2011.08.004. URL: <https://www.sciencedirect.com/science/article/pii/S0736584511001001>.
- [6] F. Sherwani, Muhammad Mujtaba Asad, and B.S.K.K. Ibrahim. "Collaborative Robots and Industrial Revolution 4.0 (IR 4.0)". In: *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*. IEEE, 2020. DOI: 10.1109/icetst49965.2020.9080724.
- [7] I. Iglesias, M. A. Sebastián, and J. E. Ares. "Overview of the State of Robotic Machining: Current Situation and Future Potential". In: *1877-7058* 132 (2015), pp. 911–917. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2015.12.577. URL: <https://www.sciencedirect.com/science/article/pii/S1877705815044896>.
- [8] Ya. L. Liberman and L. N. Gorbunova. "Selection of Positioning Devices for the Components of CNC Machines". In: *Russian Engineering Research* 41.11 (2021), pp. 1067–1070. ISSN: 1934-8088. DOI: 10.3103/S1068798X21110204. URL: <https://link.springer.com/article/10.3103/s1068798x21110204>.
- [9] Aude Billard and Danica Kragic. "Trends and challenges in robot manipulation". In: *Science* 364.6446 (2019). DOI: 10.1126/science.aat8414.
- [10] Michele Gadaleta, Marcello Pellicciari, and Giovanni Berselli. "Optimization of the energy consumption of industrial robots for automatic code generation". In: *Robotics and Computer-Integrated Manufacturing* 57 (2019), pp. 452–464. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2018.12.020. URL: <https://www.sciencedirect.com/science/article/pii/S0736584518301856>.
- [11] Panagiotis Kyratsis, Konstantinos Kakoulis, and Angelos P. Markopoulos. "Advances in CAD/CAM/CAE

- Technologies". In: *Machines* 8.1 (2020), p. 13. ISSN: 2075-1702. DOI: 10.3390/machines8010013. URL: <https://www.mdpi.com/2075-1702/8/1/13/html>.
- [12] C. K. Maiti. *Introducing technology computer-aided design (TCAD): Fundamentals, simulations and applications*. Temasek Boulevard, Singapore: Pan Stanford Publishing, 2017. ISBN: 9789814745529.
- [13] Eckart Uhlmann, Sascha Reinkober, and Tobias Hollerbach. "Energy Efficient Usage of Industrial Robots for Machining Processes". In: 2212-8271 48 (2016), pp. 206–211. ISSN: 2212-8271. DOI: 10.1016/j.procir.2016.03.241. URL: <https://www.sciencedirect.com/science/article/pii/S2212827116305303>.
- [14] Zuha Anjum et al. "Design and Modeling of 9 Degrees of Freedom Redundant Robotic Manipulator". In: *Journal of Robotics and Control (JRC)* 3.6 (2022), pp. 800–808. ISSN: 2715-5056. DOI: 10.18196/jrc.v3i6.15958. URL: <https://journal.umy.ac.id/index.php/jrc/article/view/15958>.
- [15] Paolo Boscariol et al. "Energy Optimization of Functionally Redundant Robots through Motion Design". In: *Applied Sciences* 10.9 (2020), p. 3022. DOI: 10.3390/app10093022.
- [16] Manja V. Kiréanski and Tatjana M. Petrović. "Combined Analytical- Pseudoinverse Inverse Kinematic Solution for Simple Redundant Manipulators and Singularity Avoidance". In: *The International Journal of Robotics Research* 12.2 (1993), pp. 188–196. ISSN: 0278-3649. DOI: 10.1177/027836499301200207.
- [17] Linwei Wang et al. "Optimization of redundant degree of freedom in robotic milling considering chatter stability". In: *The International Journal of Advanced Manufacturing Technology* 121.11-12 (2022), pp. 8379–8394. ISSN: 1433-3015. DOI: 10.1007/s00170-022-09889-4.
- [18] D. I. Malyshev et al. "Analysis of the Singularities Influence on the Forward Kinematics Solution and the Geometry of the Workspace of the Gough-Stewart Platform". In: *Advances in Service and Industrial Robotics*. Ed. by Andreas Müller and Mathias Brandstötter. Vol. 120. Mechanisms and Machine Science. Cham: Springer International Publishing and Imprint: Springer, 2022, pp. 60–67. ISBN: 978-3-031-04869-2. DOI: 10.1007/978-3-031-04870-8₈.
- [19] Hang Zhao et al. "Singularity Analysis and Singularity Avoidance Trajectory Planning for Industrial Robots". In: *2021 China Automation Congress (CAC)*. IEEE, 2021. DOI: 10.1109/cac53003.2021.9727497.
- [20] Paul Milenkovic. "Wrist singularity avoidance with a robot end-effector adding an oblique, redundant axis". In: *Mechanism and Machine Theory* 162 (2021), p. 104355. ISSN: 0094-114X. DOI: 10.1016/j.mechmachtheory.2021.104355. URL: <https://www.sciencedirect.com/science/article/pii/S0094114X21001130>.
- [21] Carlos Faria et al. "Position-based kinematics for 7-DoF serial manipulators with global configuration control, joint limit and singularity avoidance". In: *Mechanism and Machine Theory* 121 (2018), pp. 317–334. ISSN: 0094-114X. DOI: 10.1016/j.mechmachtheory.2017.10.025. URL: <https://www.sciencedirect.com/science/article/pii/S0094114X17306559>.
- [22] Xuan Bien Duong. "On the Effect of the End-effector Point Trajectory on the Joint Jerk of the Redundant Manipulators". In: *Journal of Applied and Computational Mechanics* 7.3 (2021), pp. 1575–1582. ISSN: 2383-4536. DOI: 10.22055/jacm.2021.35350.2635. URL: https://jacm.scu.ac.ir/article_16660.html.
- [23] R.V. Dubey, J.A. Euler, and S.M. Babcock. *Robotics and Automation, 5th IEEE International Conference on, 1988: Proceedings*. Los Alamitos: IEEE Computer Society Press, 1988. ISBN: 0818608528. URL: <http://ieeexplore.ieee.org/servlet/opac?punumber=202>.
- [24] Hoai Nam Huynh et al. "Modelling the dynamics of industrial robots for milling operations". In: *Robotics and Computer-Integrated Manufacturing* 61 (2020), p. 101852. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2019.101852. URL: <https://www.sciencedirect.com/science/article/pii/S0736584519301784>.
- [25] Chen-Gang et al. "Review on kinematics calibration technology of serial robots". In: *International Journal of Precision Engineering and Manufacturing* 15.8 (2014), pp. 1759–1774. ISSN: 2005-4602. DOI: 10.1007/s12541-014-0528-1. URL: <https://link.springer.com/article/10.1007/s12541-014-0528-1>.
- [26] A. Gasparetto and V. Zanotto. "Optimal trajectory planning for industrial robots". In: *Advances in Engineering Software* 41.4 (2010), pp. 548–556. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2009.11.001. URL: <https://www.sciencedirect.com/science/article/pii/S0965997809002464>.
- [27] Mike Vande Weghe, Dave Ferguson, and Siddhartha S. Srinivasa. "Randomized path planning for redundant manipulators without inverse kinematics". In: *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007. DOI: 10.1109/ichr.2007.4813913.
- [28] Alexander Reiter, Andreas Muller, and Hubert Gattringer. "Inverse kinematics in minimum-time trajectory planning for kinematically redundant manipulators". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016. DOI: 10.1109/iecon.2016.7793436.
- [29] K. Koray Ayten, M. Necip Sahinkaya, and Ahmet Dumlu. "Optimum Trajectory Generation for Redundant/Hyper-Redundant Manipulators". In: *IFAC-PapersOnLine* 49.21 (2016), pp. 493–500. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2016.10.651. URL: <https://www.sciencedirect.com/science/article/pii/S2405896316322637>.
- [30] Xiaohua Shi et al. "Kinematics and Singularity Analysis of a 7-DOF Redundant Manipulator". In: *Sensors* 21.21 (2021), p. 7257. ISSN: 1424-8220. DOI: 10.3390/

s21217257. URL: <https://www.mdpi.com/1424-8220/21/21/7257>.

- [31] Yanhui Wei et al. “General approach for inverse kinematics of nR robots”. In: *Mechanism and Machine Theory* 75 (2014), pp. 97–106. ISSN: 0094-114X. DOI: 10.1016/j.mechmachtheory.2014.01.008.
- [32] R. Stevenson, B. Shirinzadeh, and G. Alici. “Singularity avoidance and aspect maintenance in redundant manipulators”. In: (2002), pp. 857–862. DOI: 10.1109/ICARCV.2002.1238536. URL: <https://ieeexplore.ieee.org/abstract/document/1238536>.
- [33] N. S. Bedrossian. “Classification of singular configurations for redundant manipulators”. In: *Proceedings, IEEE International Conference on Robotics and Automation*. New York: IEEE, 2002, pp. 818–823. ISBN: 0-8186-9061-5. DOI: 10.1109/ROBOT.1990.126089.
- [34] Liguang Huo and Luc Baron. “The joint-limits and singularity avoidance in robotic welding”. In: *Industrial Robot: An International Journal* 35.5 (2008), pp. 456–464. DOI: 10.1108/01439910810893626. URL: <https://www.emerald.com/insight/content/doi/10.1108/01439910810893626/full/pdf>.
- [35] Chengkai Dai et al. “Planning Jerk-Optimized Trajectory With Discrete Time Constraints for Redundant Robots”. In: *IEEE Transactions on Automation Science and Engineering* 17.4 (2020), pp. 1711–1724. ISSN: 1545-5955. DOI: 10.1109/TASE.2020.2974771.
- [36] Li Jiang et al. “Time-Jerk Optimal Trajectory Planning for a 7-DOF Redundant Robot Using the Sequential Quadratic Programming Method”. In: 2017 (2017), pp. 343–353. DOI: 10.1007/978-3-319-65298-6_textunderscore32.
- [37] Kai Wu et al. “Review of Industrial Robot Stiffness Identification and Modelling”. In: *Applied Sciences* 12.17 (2022), p. 8719. DOI: 10.3390/app12178719. URL: <https://www.mdpi.com/2076-3417/12/17/8719>.
- [38] Gang Xiong, Ye Ding, and LiMin Zhu. “Stiffness-based pose optimization of an industrial robot for five-axis milling”. In: *Robotics and Computer-Integrated Manufacturing* 55 (2019), pp. 19–28. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2018.07.001. URL: <https://www.sciencedirect.com/science/article/pii/S0736584517304556>.
- [39] Toni Cvitanic, Vinh Nguyen, and Shreyes N. Melkote. “Pose optimization in robotic machining using static and dynamic stiffness models”. In: *Robotics and Computer-Integrated Manufacturing* 66 (2020), p. 101992. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2020.101992. URL: <https://www.sciencedirect.com/science/article/pii/S0736584520302039>.
- [40] Paryanto et al. “Reducing the energy consumption of industrial robots in manufacturing systems”. In: *The International Journal of Advanced Manufacturing Technology* 78.5-8 (2015), pp. 1315–1328. ISSN: 1433-3015. DOI: 10.1007/s00170-014-6737-z. URL: <https://link.springer.com/article/10.1007/s00170-014-6737-z>.