Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 96 (2021) 324–329

www.elsevier.com/locate/procedia

CIRPe 2020 – 8th CIRP Global Web Conference – Flexible Mass Customisation

Time-optimal Path Planning of Multi-axis CNC Processes Using Variability of Orientation

Frederik Wulle*, Max Richter, Christoph Hinze, Alexander Verl

Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW), University of Stuttgart, Seidenstrasse 36, 70174 Stuttgart, Germany

Abstract

In multi-axis processes, such as welding or 3D printing, the determination of the orientation of the tool relative to the workpiece is usually determined by technology-dependent geometric algorithms. In the simplest case, the orientation of the tool is defined according to the surface normal vectors of the workpiece surfaces. However, with large curvature gradients, this method can lead to jumps in the orientation which lead to unwanted stopping of the translational motion. In the linear interpolation of this movement, the path speed usually decreases significantly due to the relatively large compensating movement of a kinematic system. This type of path movement usually has negative effects on the CNC process, which requires constant path velocities.

This paper describes a method for time-optimized path planning that utilizes the orientation tolerances of multi-axis processes. Orientation tolerances are process-dependent tolerances that indicate which orientation ranges (angular positions) from tool to workpiece are permissible without decreasing the process quality. The potential time savings of this method are shown with an extreme example of a planar 3-axis kinematics and on an industrial 7-axis CNC system.

Keywords: Dynamics; Additive Manufacturing; Machine tool; Optimization; Tool path

1. Introduction

Computerized Numerical Control (CNC) multi-axis processing, i.e. processes with additional rotational axes beside 3 translational axes, are widely used in the production industry to produce complex components. Starting from a Computer Aided Design (CAD) process in which the workpiece's geometry is defined and afterwards in the Computer Aided Manufacturing (CAM) step, the paths of the process are planed. The programmed path is defined along a certain contour and described in a machine program (e.g. G-code). In multi-axis processes the orientation planning of a tool relative to the workpiece is a challenging task which addresses a couple of aspects such as steady path transitions or collisions. The simplest way to handle this task is the orthogonal orientation based on the workpiece's surface. For additive processes or ball milling the process dependent variability of orientation can be used for faster and smoother paths as described in detail hereafter.

2. State of the Art

Path planning strategies and optimization approaches are widely spread in science and technology. This paper addresses no point-to-point motion formalism but continuous path formulations. The orientation of a tool relative to a workpiece can be seen as a set of degrees of freedom that can be used to achieve different goals in path planning. In case of milling, usually the tool orientation has a great influence on the resulting workpiece geometry. Therefore, the tool orientation can be optimized in order to minimize geometrical errors [2]. However, for some technologies, the exact tool orientation is of lower importance. For technologies like 3-D printing and welding the aforementioned degrees of freedom can be used to plan time-optimal paths, what is to be covered in this work.

A common problem of time-optimal path planning is to move the Tool Center Point (TCP) along an edge or a corner of the workpiece. In such situations, the position of the TCP as a function of time is not a steadily differentiable curve. There-

* Corresponding author. Tel.: +49-711-685-8407.
E-mail address: Frederik.Wulle@isw.uni-stuttgart.de (FrederikWulle).

10.1016/j.procir.2021.01.095

fore, the speed would have to change infinitely fast when the edge is passed at a non-zero speed. This would require an infinitely great acceleration and jerk, which would come with infinitely great forces as well and thus cannot be realized on a physical system. There are two options to solve this problem. One possibility is just to stop at the edge during the reorientation of the tool and then accelerate again. This approach grants an exact following of the path but also slows down the process time significantly. The other option is to leave the programmed path in order to obtain a smooth curve for the position of the tool.

There are two major ideas to obtain smooth motions at the corners of the workpiece by leaving the ideal path of the TCP. The first one is smoothing at the corners. Here, a section around the corner is defined on which the ideal path is replaced by a path section that enables all axes to follow curves that are several times steady differentiable. Often, polynomials are chosen for these axis profiles [4]. The second approach is based on leaving the ideal path not only at the corner but during almost the entire path and is called tube following [3]. The name originates from the idea that the TCP has to stay within a tube surrounding the programmed path. As long as it does not leave this tube the resulting degrees of freedom can be used for deriving smooth and time optimal paths. To use the orientation tolerances for faster processing is poorly researched so far.

3. Concept for time-optimal path planning

In this chapter the core concept is introduced. First, two possibilities of suitable definition for orientation tolerances are described. Then, an established optimization algorithm is extended to the use in context of orientation tolerances.

3.1. Definition of orientation tolerances

For certain technologies a continuum of feasible orientations between the tool and the workpiece exists. Orientation tolerances can be used to describe the resulting set of admissible orientations. Fig. 1 shows the tool in a coordinate system whose origin is in the TCP. The U-axis is orthogonal to the workpiece's surface, while the V-axis is tangential to the path. The W-axis completes the right-handed coordinate system. Δ_1 is the pitch angle and Δ_2 the roll angle of the tool. If both angles stay within their respective constraints, according to

$$\underline{\mathbf{\Delta}}(s) \leq \mathbf{\Delta}(s) \leq \overline{\mathbf{\Delta}}(s) , \tag{1}$$

the tool stays within a pyramid.

In cases where both angles, the roll and the pitch angle, are simultaneously high, the process quality could decrease. Here, it might be advantageous to keep the tool inside of a cone (see Fig. 2). The corresponding inequality is

$$(\alpha_1 - \Delta_1)^2/k_{\Delta 1}^2 + (\alpha_2 - \Delta_2)^2/k_{\Delta 2}^2 \leq 1 \tag{2}$$

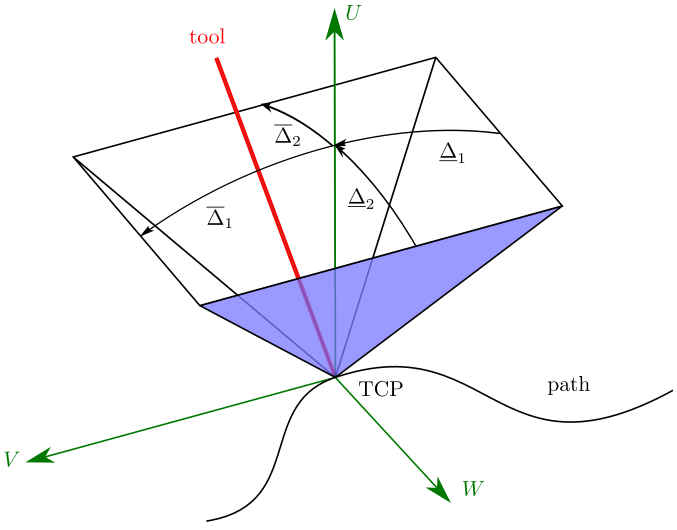

Fig. 1: Axes-wise tolerances of admissible tool deviation.

where α_1 and α_2 define the permitted range, of the pitch and the roll angle, respectively. The parameters $k_{\Delta 1}$ and $k_{\Delta 2}$ can be used to vary the allowed deviation in the V- and W-axis.

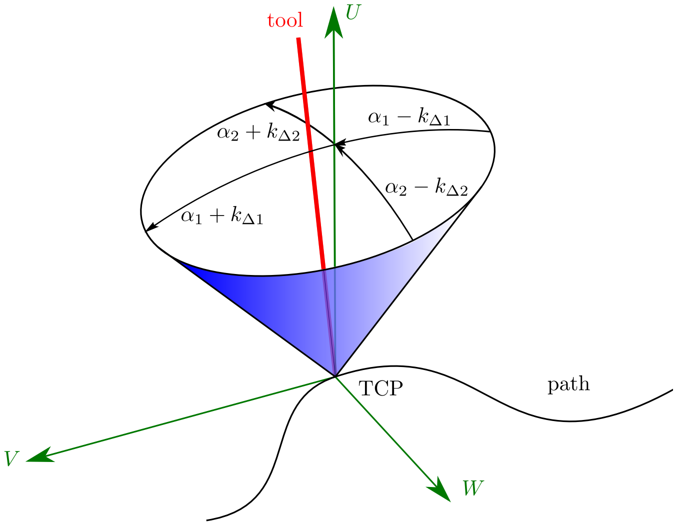
tool
U
$\alpha_1 - k_{\Delta 1}$
$\alpha_2 + k_{\Delta 2}$
$\alpha_2 - k_{\Delta 2}$
$\alpha_1 + k_{\Delta 1}$
TCP
path
V
W

Fig. 2: Cone-wise tolerances of admissible tool deviation.

Both definitions of orientation tolerance are later used in the optimization formulation.

3.2. Time-optimal path planning algorithm

The foundation of the approach described in this paper is the convex optimization approach for time-optimal path planning given in [6] and extended by allowing tool orientation deviations $\mathbf{\Delta}$ in Cartesian space. In this publication, a mechanical system with known equations of motion is studied. The problem

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

is set in a way, that the positions of the axes $\boldsymbol{q}$ are described as functions of the path coordinate s which decouples the problem of finding the path $\boldsymbol{q}(s)$ from the dynamics planning problem, i.e. finding a suitable trajectory $s(t)$. In the special case that $\boldsymbol{q}$ is normalized in path length, $s(t)$ directly corresponds to the linear feed velocity, i.e. $s(t) = v_f(t)$. From this follows the composite function

$$\boldsymbol{q} = \boldsymbol{q}\big(s(t)\big)\,. \tag{3}$$

For a given geometrical path s is used to describe on which point of the path the TCP is located. As the TCP moves along the path from start point to end point, the value of s changes from zero to one. Thus, $s(t)$ carries the information on how fast each point on the path is passed. $s(t)$ is to be optimized while the geometrical path of the tool is predefined by $\boldsymbol{q}(s)$. This yields the optimization problem

$$\begin{aligned}
\min_{T,s(\cdot),\boldsymbol{\tau}(\cdot)} \quad & T && \text{(4a)}\\
\text{s. t.} \quad & \boldsymbol{\tau}(t) = \boldsymbol{m}(s(t))\,\ddot{s}(t) + \boldsymbol{c}(s(t))\,\dot{s}(t)^2 + \boldsymbol{g}(s(t)) && \text{(4b)}\\
& s(0) = 0\,, \quad s(T) = 1 && \text{(4c)}\\
& \dot{s}(0) = \dot{s}_0\,, \quad \dot{s}(T) = \dot{s}_T\,, \quad \dot{s}(t) \geq 0 && \text{(4d)}\\
& \underline{\boldsymbol{\tau}}(s(t)) \leq \boldsymbol{\tau}(t) \leq \overline{\boldsymbol{\tau}}(s(t)) && \text{(4e)}\\
& \text{for } t \in [0, T]\,.
\end{aligned}$$

The goal is to minimize the total processing time T, by applying forces and torques $\boldsymbol{\tau}$ at each time step t. As the spatial trajectory $\boldsymbol{q}(s)$ is predefined, the equations of motion (4b) are rearranged to only depend on the unknown path parameter $s(t)$, which is subject to optimization (cf. [6]). $\boldsymbol{m}$ is the generalized mass vector while $\boldsymbol{c}$ and $\boldsymbol{g}$ are the vector of generalized Coriolis and centrifugal forces and the vector of generalized gravitational forces, respectively. In [6] a derivation of these variables can be found. Due to inequality (4e), the constraints on the positioning forces and positioning torques limit the attainable processing time. The boundary conditions (4c) – (4d) ensure, that the position and speed of the tool take on certain values at the begin and the end of the considered path. Since for most applications the tool stands still before and after the process, $\dot{s}$ can be assumed to zero at the boundaries. Via transformation

$$T = \int_0^T 1\,\mathrm{d}t = \int_{s(0)}^{s(T)} \frac{1}{\dot{s}}\,\mathrm{d}s = \int_0^1 \frac{1}{\dot{s}}\,ds \tag{5}$$

and the substitutions $a(s) = \ddot{s}$ and $b(s) = \dot{s}^2$ the problem can be transcribed into the form

$$\begin{aligned}
\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot)} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}}\,\mathrm{d}s && \text{(6a)}\\
\text{s. t.} \quad & \boldsymbol{\tau}(s) = \boldsymbol{m}(s)\,a(s) + \boldsymbol{c}(s)\,b(s) + \boldsymbol{g}(s) && \text{(6b)}\\
& b(0) = \dot{s}_0^2\,, \quad b(1) = \dot{s}_T^2 && \text{(6c)}\\
& b'(s) = 2a(s) && \text{(6d)}\\
& b(s) \geq 0 && \text{(6e)}\\
& \underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}(s) && \text{(6f)}\\
& \text{for } s \in [0, 1]\,.
\end{aligned}$$

As a result of the transformation, the dynamics equation (6b) depends linearly on the decision variables and it can be shown that the problem is convex which is beneficial for its numerical solution. Equation (6d) reflects the fact, that $\dot{s}$ and $\ddot{s}$ as derivations of s cannot be chosen independently. Inequality (6e) prevents $\dot{s}$ from taking imaginary values.

Solving problem (6) provides the time-optimal way to move the axes along predefined profiles. However, this solution does not provide any information on how to chose these profiles in order to be able to achieve the minimal processing time. When it comes to taking advantage of orientation tolerances, $\boldsymbol{q}(s)$ becomes subject to optimization, because a change of the tool orientation also causes a change of the axis configuration while the TCP path remains the same. This leads to a new optimization problem, whose objective is to choose an optimal tool orientation at any point on the path, by minimal processing time.

To integrate orientation tolerances $\boldsymbol{\Delta}(s(t))$ into the optimization problem (6), the inverse kinematics transformation $\boldsymbol{\phi}^{-1}(\cdot)$ from workpiece surface coordinates $\boldsymbol{y}$ to axis coordinates $\boldsymbol{q}$ is required, i.e. $\boldsymbol{q}(s) = \boldsymbol{\phi}^{-1}(\boldsymbol{y}(s), \boldsymbol{\Delta}(s))$. This allows to pass a predefined trajectory $\boldsymbol{y}(s) = \bar{\boldsymbol{y}}(\boldsymbol{s})$ (instead of a predefined $\boldsymbol{q}(s)$) and to add the orientation tolerances $\boldsymbol{\Delta}(s)$ as optimization variables, which results in the optimization problem

$$\begin{aligned}
\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot),\boldsymbol{\Delta}(\cdot)} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}}\,\mathrm{d}s && \text{(7a)}\\
\text{s.t.} \quad & \boldsymbol{\tau}(s) = \boldsymbol{m}(\boldsymbol{\Delta}(s))a(s) + \boldsymbol{c}(\boldsymbol{\Delta}(s))\boldsymbol{b}(s) + \boldsymbol{g}(\boldsymbol{\Delta}(s)) && \text{(7b)}\\
& s(0) = 0\,, \quad s(T) = 1 && \text{(7c)}\\
& \dot{s}(0) = \dot{s}_0\,, \quad \dot{s}(T) = \dot{s}_T\,, \quad \dot{s}(t) \geq 0 && \text{(7d)}\\
& \boldsymbol{\Delta}(s(0)) = \boldsymbol{\Delta}_0, \quad \boldsymbol{\Delta}(s(T)) = \boldsymbol{\Delta}_T && \text{(7e)}\\
& \boldsymbol{\Delta}'(s(0)) = \boldsymbol{\Delta}'_0, \quad \boldsymbol{\Delta}'(s(T)) = \boldsymbol{\Delta}'_T && \text{(7f)}\\
& \underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}(s) && \text{(7g)}\\
& \underline{\boldsymbol{\Delta}}(s) \leq \boldsymbol{\Delta}(s) \leq \overline{\boldsymbol{\Delta}}(s) && \text{(7h)}\\
& \underline{\boldsymbol{\Delta}}'(s) \leq \boldsymbol{\Delta}'(s) \leq \overline{\boldsymbol{\Delta}}'(s) && \text{(7i)}\\
& \text{for } s \in [0, 1]\,.
\end{aligned}$$

Usually, the tool orientation as well as the rate of change of the tool orientation at the beginning and end of the path cannot be freely chosen. This is taken into account by the new boundary constraints (7e) – (7f). (7h) constrains the maximum deviation of tool orientation, as described in (1), but can also be changed to other constraints, e.g. in (2). Constraint (7i) avoids changes of the tool orientation, that are fast enough to disrupt the process.

Now that orientation tolerances are used to optimize the path, the axis positions $\boldsymbol{q}$ vary with the tool orientation and, hence, depending on the chosen tool orientation. In contrast to equation (6b), $\boldsymbol{m}$, $\boldsymbol{c}$ and $\boldsymbol{g}$ are functions of $\boldsymbol{\Delta}$ and also depending on the tool predefined tool trajectory $\bar{\boldsymbol{y}}$. Therefore, the dynamics equation (7b) is no longer linear in the optimization variables and its convexity depends on the local convexity of the inverse kinematics $\boldsymbol{\phi}^{-1}(\cdot)$. Note, that this can be the case for simple machine tool kinematics and can in general be achieved for nonlinear kinematics via semidefinite, convex relaxations [8]. But even without the convexity a feasible local solution is often found improving the total processing time.

4. Implementation and Validation

The presented algorithm is implemented and then validated, first, in a an academic simulation example and second, in an industrial multi-axis CNC machine tool.

4.1. Implementation of optimization algorithm

The implementation of the algorithm for solving problem (7) is implemented in MATLAB. The mechanical model of machines is delivered by a MATLAB live script using the Symbolic Math Toolbox [5]. The optimization problem is then solved by nonlinear programming. For this, `CasADi` [1] is used with the IPOPT solver [7]. The algorithm was implemented in a modular way and so different machine configurations and numbers of axes can be handled.

4.2. Simulation of a plane kinematics

In order to test the solution for plausibility and validate the path planning optimization, the simulation of a simple system is analysed. Fig. 3 shows a schematic system as a plane 3-axis kinematics. Masses m_1 and m_2 are assigned to the two rigid bodies. S_1 and S_2 mark the centers of mass. For the rotating body d_{S2}, the distance from S_2 to the rotation point, the length of the entire body d and the moment of inertia J_2 are given. The state of the system is defined by the axes $\boldsymbol{q} = (q_1,\ q_2,\ q_3)^T$, while the orientation of the tool is described by Θ. The TCP is marked red. On the right side of the kinematics a workpiece with a 90° corner is shown. n_1 and n_2 are normal vectors of the workpiece's surface in the points P_1 and P_2, respectively. The dynamics of the system can be written in the form

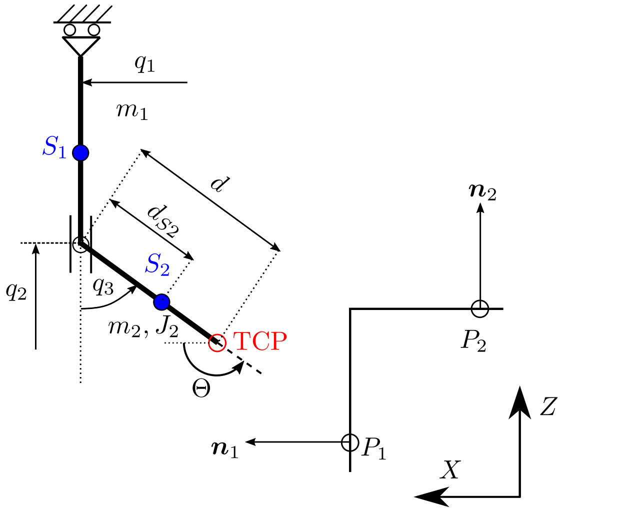

Fig. 3: Schematic plane 3-axis kinematics.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q})\ \text{, where} \tag{8}$$

$$\boldsymbol{\tau} = \begin{pmatrix} F_1 & F_2 & T_3 \end{pmatrix}^T, \tag{9}$$

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} m_1 + m_2 & 0 & -m_2 d_{S2}\cos(q_3) \\ 0 & m_2 & m_2 d_{S2}\sin(q_3) \\ -m_2 d_{S2}\cos(q_3) & m_2 d_{S2}\sin(q_3) & J_2 \end{pmatrix}, \tag{10}$$

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{pmatrix} 0 & 0 & m_2 d_{S2}\sin(q_3)\dot{q}_3 \\ 0 & 0 & m_2 d_{S2}\cos(q_3)\dot{q}_3 \\ 0 & 0 & 0 \end{pmatrix} \text{ and}$$

$$\boldsymbol{G}(\boldsymbol{q}) = \begin{pmatrix} 0 & m_2 g & m_2 d_{S2}\sin(q_3) g \end{pmatrix}^T. \tag{11}$$

$\boldsymbol{\tau}$ consists of the positioning forces and torques in each one of the 3-axis directions. Therefore, each column of the dynamics equation (8) describes the dynamics in the direction of one axis. g in (11) is the acceleration factor of gravity. The matrices $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ can be transformed by the path description $\boldsymbol{q}(s)$ into $\boldsymbol{m}$, $\boldsymbol{c}$ and $\boldsymbol{g}$.

To validate the algorithm three different approaches for moving the TCP of the plane 3-axis kinematics along a predefined path are simulated. The task is to follow the surface of the workpiece from point $P1$ to point $P2$. The first approach keeps the tool orthogonal to the workpiece surface and turns the tool at the corner before moving on. This approach shall be called the orthogonal approach. In the second example the tool is turned linearly for 45° while the TCP moves from $P1$ to the corner and then again 45° while moving from the corner to $P2$. This one shall be called the linear approach. The third and last approach to be tested is the optimized path, delivered by the new opti-

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

mization algorithm. Fig. 4 shows the optimized solution for the task.

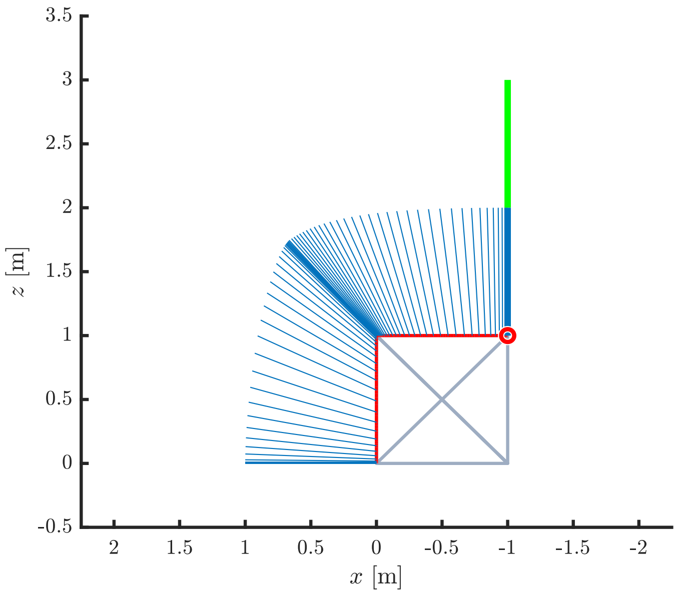

Fig. 4: Time optimal path for the plane 3-axis kinematics (blue lines are time step constant).

The results of the validation can be seen in table 1. The simulation is executed for different maximum torques between the two bodies of the kinematics. It can be seen, that the linear approach is generally faster than the orthogonal approach but the optimized path allows even shorter processing times. Another result of the experiment is the perception that the percentage of time that can be saved depends on the physical parameters.

Table 1: Processing times for different approaches and parameters.

	1 Nm	10 Nm	100 Nm
Orthogonal	4.120 s	3.551 s	3.525 s
Linear	0.775 s	0.563 s	0.466 s
Optimized	0.617 s	0.477 s	0.432 s

4.3. Validation on CNC multi-axis machine

The trajectories have been validated in a simulation. In the following, a validation on a real machine tool in performed. For this purpose a G-code is automatically generated in MATLAB by extraction of sampled axes set points. Fig. 5 shows a picture of the 7-axis machine tool with all coordinate systems that are used in order to describe the kinematics. Two kinematic chains expand from the inertial machine system M. One kinematic chain connects the axes that actuate the tool and ends at the TCP. The other one goes over the axes that move the machine table and ends at the workpiece's coordinate system. Both kinematic chains are represented by blue dashed lines. The red and yellow coordinate systems show how the machine can move. Since there are seven axes between the table and the tool, the machine has kinematic redundancy.

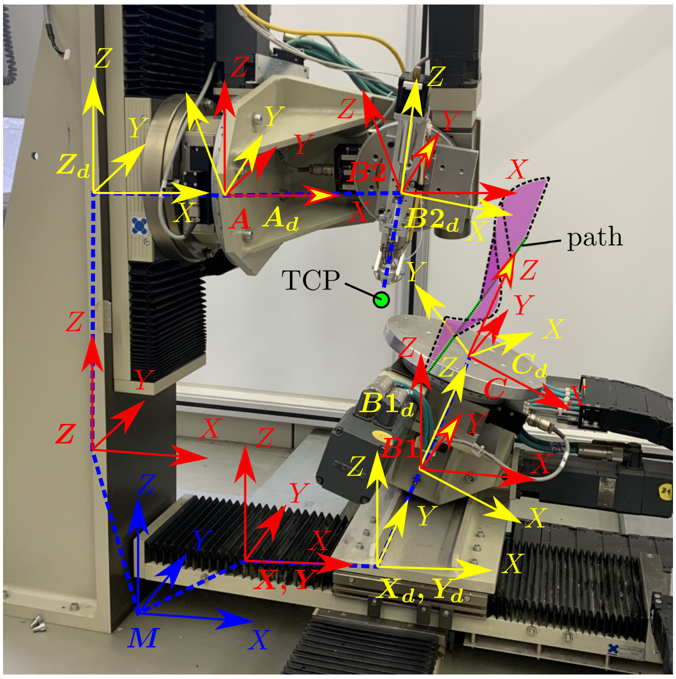

Fig. 5: Machine tool used for the validation.

The generated G-code is decoded and executed by a TwinCAT 3.1 control system from Beckhoff which reconstructs the path online and interpolates it in the real-time-cycle frequency of 1 kHz. The defined sample axes set points are interpolated so that the CNC is forced to execute the planned path with a given feedrate. The path is chosen for testing the algorithm leads the TCP on a straight line relative to the machine table coordinate system C_d. The normal vector of the fictive workpiece's surface changes sinusoidally. So the resulting path lies on a fictional workpiece that is twisted in itself. In Fig. 5 the straight line on which the TCP moves along can be seen as a green line with black dots. The orientations that the tool takes, when moving orthogonal to the workpiece are outlined with dashed lines and a translucent purple surface. To maintain a clear overview the TCP is not shown on the path.

Fig. 6 illustrates how the optimization uses the orientation tolerance to avoid as much movement of the axes as possible. For the non-optimized case, both $B1$-axis and $B2$-axis have to be used, to keep the tool orthogonal to the workpiece surface for the entire path. The optimized solution makes use of the more dynamic $B2$-axis, instead of rotating the much heavier and therefore much slower $B1$-axis of the table. The $B2$-axis is only moved as far as necessary in order not to violate the orientation tolerance that has been defined. If the provided orientation tolerance is decreased at some point the optimized solution will include movement of the $B1$-axis as well, due to the $B2$-axis limits. As a result of the optimization the process can be completed in 2.13 s instead of 3.72 s, which corresponds to a reduction by about 43%.

Fig. 7 compares the feedrates of the optimized and the non-optimized cases. When staying orthogonal to the workpiece surface the machine has to slow down at some parts of the path.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

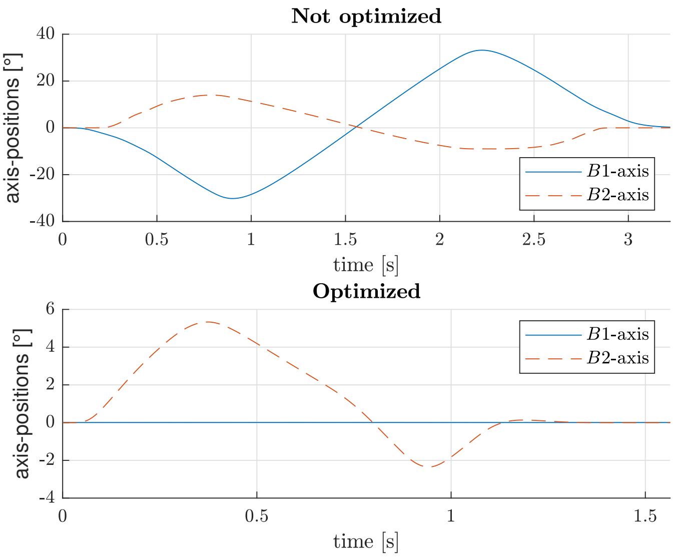

Fig. 6: Two axes over time for non-optimized and optimized case (note the different scaling of abscissas).

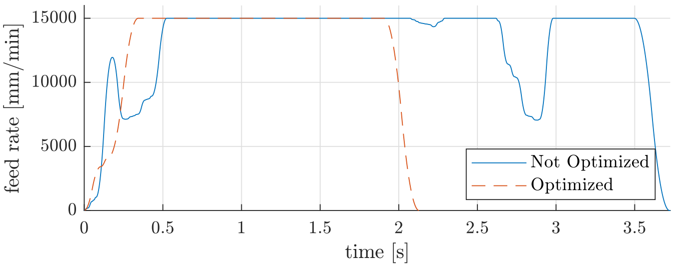

Fig. 7: Comparison of the feed rates.

The optimization enables the machine to maintain a constant velocity for the entire path, except during the slopes at the beginning and the end of the path. Here, the feedrate refers to the main translational axes (X, Y and Z). Due to the larger rotational angles in the non-optimized case, larger alignment movements must be performed by the kinematic transformation. Thus, the total path length of the optimized solution is significantly shorter.

Fig. 8 shows the angle deviations relative to the path coordinate system for the task previously described. The tool has to stay within a pyramid like the one in Fig. 1. The results for $-\underline{\boldsymbol{\Delta}}(s) = \overline{\boldsymbol{\Delta}}(s) = [15^\circ, 15^\circ]^\mathrm{T}$ and for both angles set to 46° can be seen. The limits are illustrated as black dotted lines. With 46° the optimal solution for the roll angle is a perfect sinusoidal curve with an amplitude of 45°. This resembles the tool staying upright while the path coordinate system rolls with the twisted surface of the workpiece, like it is outlined in Fig. 5. The pitch angle moves in a narrow range around zero, allowing for smooth acceleration a deceleration. When restricting both of the angles to 15° the roll angle reaches its limits at the path pieces where the workpiece is turned most. For this case the optimization uses the pitch angle more for obtaining optimal acceleration profiles despite the restriction of the roll angle.

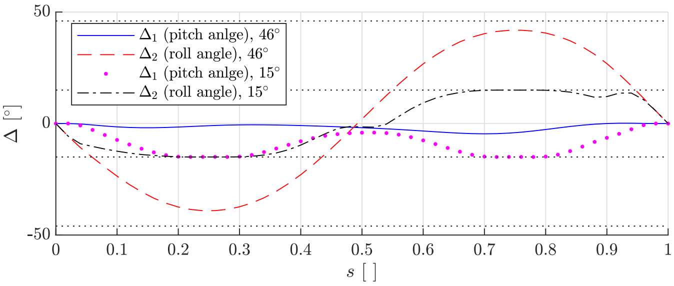

Fig. 8: Comparison of the angle deviations for different limits.

5. Conclusion and Future Work

A method for time-optimal path planning of multi-axis processes using variability of orientation is presented, in which first the orientation tolerances are formulated and then the optimization algorithm was defined and experimentally tested. Reductions in execution time can be achieved through sufficiently large tolerances. A practical validation on a real workpiece remains future work. Additionally, consideration of collision avoidance and the real-time implementation of this algorithm into a CNC control are further research topics.

6. Acknowledgements

The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationsCampus “Mobilität der Zukunft” and DFG for partial support under Germany’s Excellence Strategy EXC 2120/1 390831618.

References

[1] Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. Casadi: a software framework for nonlinear optimization and optimal control. Mathematical Programming Computation 11, 1–36. doi:10.1007/s12532-018-0139-4.

[2] Chiou, J.C.J., Lee, Y.S., 2005. Optimal tool orientation for five-axis tool-end machining by swept envelope approach. Journal of Manufacturing Science and Engineering 127, 810–818. doi:10.1115/1.2035698.

[3] Debrouwere, F., van Loock, W., Pipeleers, G., Swevers, J., 2014. Time-optimal tube following for robotic manipulators, in: 2014 IEEE 13th International Workshop on Advanced Motion Control (AMC), IEEE, Piscataway, NJ. pp. 392–397. doi:10.1109/AMC.2014.6823314.

[4] Olomski, J., 1989. Bahnplanung und Bahnführung von Industrierobotern: Zugl.: Braunschweig, Univ., Diss. volume 4 of *Fortschritte der Robotik*. Vieweg, Braunschweig.

[5] The MathWorks, I., 2019. Symbolic Math Toolbox. Natick, Massachusetts, United State. URL: https://www.mathworks.com/help/symbolic/.

[6] Verscheure, D., Demeulenaere, B., Swevers, J., de Schutter, J., Diehl, M., 2009. Time-optimal path tracking for robots: A convex optimization approach. IEEE Transactions on Automatic Control 54, 2318–2327. doi:10.1109/TAC.2009.2028959.

[7] Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57. doi:10.1007/s10107-004-0559-y.

[8] Yenamandra, T., Bernard, F., Wang, J., Mueller, F., Theobalt, C., 2019. Convex optimisation for inverse kinematics, in: 2019 International Conference on 3D Vision (3DV), IEEE. pp. 318–327.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.