https://doi.org/10.20965/ijat.2021.p0621

Paper:

Kinematic Tool-Path Smoothing for 6-Axis Industrial Machining Robots

Shingo Tajima†, Satoshi Iwamoto, and Hayato Yoshioka

Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan
†Corresponding author, E-mail: tajima.s.ac@m.titech.ac.jp
[Received February 25, 2021; accepted April 27, 2021]

The demands for machining by industrial robots have been increasing owing to their low installation cost and high flexibility. A novel trajectory generation algorithm for high-speed and high-accuracy machining by industrial robots is proposed in this paper. Linear interpolation in the workspace and smooth trajectory generation at the corners are important in industrial machining robots. Because industrial robots are composed of rotational joints, the joint space has a nonlinear relationship with the workspace. Therefore, linear interpolation in the joint space, which has been widely used in conventional machine tools, does not guarantee linear interpolation in the actual machining workspace. This results in the degradation of the machining surface. The proposed trajectory generation algorithm based on the decoupled approach can achieve linear interpolation in the workspace by separating the position commands into Cartesian coordinates and the orientation commands into spherical coordinates. In addition, a novel corner smoothing method that generates a smooth and continuous trajectory from discrete commands is proposed in this paper. The proposed kinematic local corner smoothing generates a smooth trajectory by using a 3-segmented constant jerk profile at the corners in the joint space. The sharp corners can thereby be replaced by smooth curves. The resulting cornering error is controlled by varying the cornering duration. The simulation results demonstrate the effectiveness of the proposed kinematic smoothing algorithm in achieving linear tool motion in straight sections and in generating smooth trajectories at corner sections within the user-defined tolerance.

Keywords: robot machining, trajectory generation, corner smoothing, industrial robots, numerical control

1. Introduction

Industrial robots are increasingly being used for machining [1–3]. The cycle time and operating costs are important issues in the machining of large workpieces [4, 5]. This is because the workspace in conventional machine tools is limited by the size of the machine. In addition, the installation of machine tools requires large expenses. In contrast, industrial robots have various advantages over machine tools such as a larger workspace, higher flexibility, and lower installation cost. Therefore, machining by industrial robots has become an alternative machining technology to machine tools [1, 2, 4, 6].

One key issue in the use of industrial robots for machining is the teaching process. Teaching is necessary to provide commands to industrial robots currently in common use. This teaching process is time-consuming and difficult to apply to actual machining processes. In contrast, machine tools are commanded by a computer-aided manufacturing (CAM) system without a teaching process. This teaching-less scheme is important for realizing industrial robot machining.

The tool-path commands generated by the CAM system are composed of a series of linear segments [7]. Because the velocity is discontinuous at a sharp corner composed of two linear segments, an infinite acceleration is required to track the tool-path. To avoid this situation, the feed-rate must be stopped at each corner. However, this causes issues in the machining quality and efficiency [8, 9]. Therefore, the generation of a smooth trajectory from these discrete commands is an important problem that needs to be solved to achieve high-speed and high-accuracy machining.

There have been numerous studies on corner smoothing for 3-axis and 5-axis machine tools [6, 10–21]. In particular, in many methods, the trajectories of rotational joints in 5-axis machine tools are smoothed in the joint coordinate system (JCS) [22, 23]. However, trajectory generation in the workpiece coordinate system (WCS) is important for realizing high-precision machining. Although linear interpolation in the workspace is also important for trajectory generation in industrial robots that consist of rotational joints, research on trajectory generation for industrial robots has mainly been performed in the JCS [24–27]. Although previous studies on trajectory generation in the WCS have also been conducted, they used optimization methods based on numerical analysis to generate linear motion trajectories [28, 29]. Because these numerical methods require a large amount of computation time, an analytical trajectory generation method is necessary for real-time systems. In addition, there is limited research

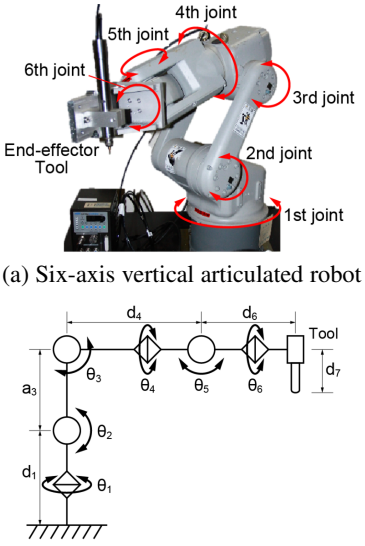


Fig. 1. Kinematic structure of the 6-axis vertical articulated robot.

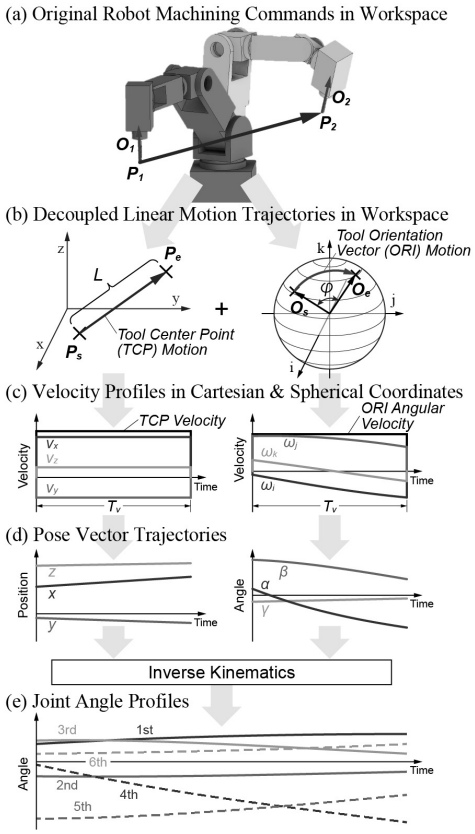


Fig. 2. Decoupled approach for trajectory generation in industrial robots.

on trajectory generation by corner smoothing for industrial robots [30].

In this study, we propose a trajectory generation method based on linear interpolation in the workspace and kinematic corner smoothing for industrial robot machining.

2. Linear Tool Pose Interpolation for Robot Machining

Figure 1 illustrates the kinematic structure of a typical 6-axis industrial robot. This type of industrial robot is called a vertical-articulated robot. This serial manipulator has six rotational joints, $\boldsymbol{q} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$, which allows it to place an end-effector on the robot in an arbitrary pose (position and orientation), $\boldsymbol{r} = [x, y, z, \alpha, \beta, \gamma]^T$, in the workspace.

Linear interpolation of the workpiece coordinates is important for machining with industrial robots. A tool motion linearly interpolated in joint coordinates becomes a fluctuating trajectory in the workpiece coordinates. This fluctuation of the tool motion in the workspace is caused by the nonlinear relationship between the joint space and the work space. Consequently, the resultant machined surface is distorted.

To realize smooth tool motion for machining, the trajectory must be linearly interpolated directly in workpiece coordinates rather than in joint space so that high-precision robot machining can be achieved using industrial robots.

2.1. Decoupled Approach for Industrial Robots

To generate a linear trajectory of the tool motion in workpiece coordinates, a decoupled approach is used to generate a smooth tool position and orientation trajectory in the workpiece coordinates. **Fig. 2** illustrates the decoupled approach for generating the trajectory of industrial robots.

Instead of generating the trajectory in joint coordinates, the decoupled approach generates a trajectory in the workspace coordinates. As shown in **Fig. 2(b)**, the tool center point (TCP) commands are interpolated in the Cartesian coordinate system, and the tool orientation vector (ORI) commands are interpolated in the spherical coordinate system. The TCP commands in Cartesian coordinates correspond to the first three components of the tool pose vector. The ORI commands in spherical coordinates are calculated using the last three components of the tool pose vector as

$$\begin{cases} i = \sin(\beta) \\ j = -\sin(\alpha)\cos(\beta) \\ k = \cos(\alpha)\cos(\beta) \end{cases} \quad \cdots\cdots\cdots \quad (1)$$

By generating the TCP and ORI trajectories in each coordinate system with the same interpolation duration T_v, as shown in **Fig. 2(c)**, synchronized tool pose motion in the WCS is realized.

Although the tool pose vector has six components, the position and orientation commands for the decoupled approach use only five components to define the TCP and ORI commands in each coordinate system. This is because only two components, α and β, are required to define the tool orientation in spherical coordinates, as shown

in Eq. (1). One component, which corresponds to the rotation around the tool orientation vector, is ignored in the spherical coordinate system. Because the generated trajectory is smooth, the remaining component, γ, is also smoothly interpolated in synchronization with the other interpolated components.

After generating the TCP and ORI trajectories, joint command trajectories are generated by converting the tool motion using inverse kinematics. Linear motion in the workspace can then be realized by sending the generated joint trajectories as commands to each servo system of the joint axis.

2.2. Linear Point-to-Point Interpolation

In the proposed decoupled approach for industrial robots, the TCP and ORI commands are interpolated in separate coordinate systems for linear motion in the workspace. To realize linear tool motion in the WCS, the TCP trajectory is generated in Cartesian coordinates. The equation for the linear interpolation (Lerp) between the position vectors $\boldsymbol{P}_1 = [x_1, y_1, z_1]^T$ and $\boldsymbol{P}_2 = [x_2, y_2, z_2]^T$ is

$$P(t) = \frac{(L - f(t)t)\,\boldsymbol{P}_1 + (f(t)t)\,\boldsymbol{P}_2}{L}, \quad \ldots\ldots \quad (2)$$

where $f(t)$ is the feedrate of the linear TCP motion, and L is the distance between $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ defined as $L = ||\boldsymbol{P}_2 - \boldsymbol{P}_1||$.

Simultaneously, the ORI command is interpolated in spherical coordinates to realize smooth angular velocity motion. To generate a trajectory in the spherical coordinate system, spherical linear interpolation (Slerp) is used instead of Lerp. The equation for Slerp between $\boldsymbol{O}_1 = [i_1, j_1, k_1]^T$ and $\boldsymbol{O}_2 = [i_2, j_2, k_2]^T$ is

$$\boldsymbol{O}(t) = \frac{\boldsymbol{O}_1 \sin(\varphi - \omega(t)t) + \boldsymbol{O}_2 \sin(\omega(t)t)}{\sin(\varphi)}, \quad . \quad (3)$$

where $\omega(t)$ is the angular velocity of the angular motion, and φ is the angle between $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ defined as $\varphi = \cos^{-1}(\boldsymbol{O}_1 \cdot \boldsymbol{O}_2)$.

Because the translational and orientational motions of the tool are synchronized, the durations of Lerp and Slerp are set to the same values. For example, for a given constant feedrate F, the duration of Lerp, T_v, is calculated using the traveling distance and feedrate as

$$T_v = \frac{L}{F} = \frac{\|\boldsymbol{P}_2 - \boldsymbol{P}_1\|}{F} \quad \ldots\ldots \quad (4)$$

Then, to realize synchronized tool motion, the angular velocity for the orientational motion is modulated as

$$\omega = \frac{\varphi}{T_v} = \frac{\cos^{-1}(\boldsymbol{O}_1 \cdot \boldsymbol{O}_2)}{T_v} \quad \ldots\ldots \quad (5)$$

based on the interpolation duration of the translational motion. Using the same interpolation duration results in synchronized TCP and ORI trajectories in the workspace.

After generating the trajectories in each coordinate, the components of the pose vector are determined. The first three components of the pose vector define the tool position, which is equivalent to the components of the TCP trajectory in Cartesian coordinates. The last three components are the rotational components, for which the components α and β are determined from the ORI trajectory in spherical coordinates as

$$\begin{cases} \alpha = \tan^{-1}\left(-\dfrac{j}{k}\right) \\ \beta = \sin^{-1}(i) \end{cases} \quad \ldots\ldots \quad (6)$$

The last component γ is interpolated in synchronization with the tangential velocity $f(t)$ and angular velocity $\omega(t)$ profiles to generate a smooth synchronized pose trajectory as follows:

$$\begin{aligned} \gamma(t) &= \frac{\gamma_1(L - f(t)t) + \gamma_2(f(t)t)}{L} \\ &= \frac{\gamma_1(\varphi - \omega(t)t) + \gamma_2(\omega(t)t)}{\varphi} \end{aligned} \quad \ldots\ldots \quad (7)$$

The trajectory of the joint commands is then derived using inverse kinematics from the generated pose-vector trajectory.

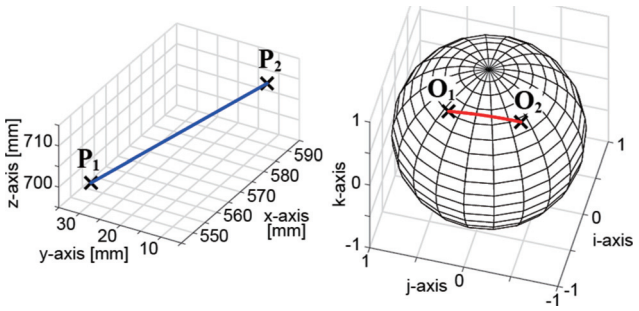
2.3. Illustrative Example

An example of a simple linear tool-path interpolation using the decoupled approach for a 6-axis industrial robot is provided in this section. **Fig. 3** shows the simulation results for the linear point-to-point tool-path interpolation in the workspace coordinates. The tool-pose vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are listed in **Table 1**. From these pose vectors, the position and orientation commands, $[\boldsymbol{P}_1, \boldsymbol{O}_1]$ and $[\boldsymbol{P}_2, \boldsymbol{O}_2]$, in each coordinate system are determined, as shown in **Fig. 3(a)**. Using Lerp and Slerp in Eqs. (2) and (3), tool trajectories are generated in each coordinate system.

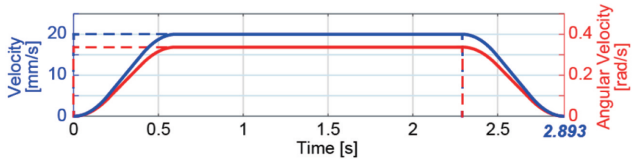
Figure 3(b) shows the tangential and angular velocities of the interpolated trajectories in each coordinate system. When the feed F and angular velocity ω are constant values in Eqs. (2) and (3), the tangential velocities of the generated trajectory become square shaped, as shown in **Fig. 3(b)** as dashed lines. This results in non-zero velocity and acceleration values at the edges of the interpolated trajectories. To generate smooth motion at both edges of the trajectories, smooth feed and angular velocity commands are used for the Lerp and Slerp calculations in Eqs. (2) and (3) instead of constant velocity and angular velocity commands. The solid lines in **Fig. 3(b)** show the generated profiles with S-shaped velocity and angular velocity commands in the workspace. As a result, smooth displacement, velocity, and acceleration profiles are generated in each coordinate system, as shown in **Fig. 3(c)**.

The trajectory of the tool pose vector is calculated from the interpolated trajectories of each coordinate, as shown in **Fig. 3(d)**. The remaining component of the command pose vector γ is interpolated in synchronization with the other generated profiles. Finally, joint kinematic profiles are calculated using inverse kinematics, as shown in **Fig. 3(e)**.

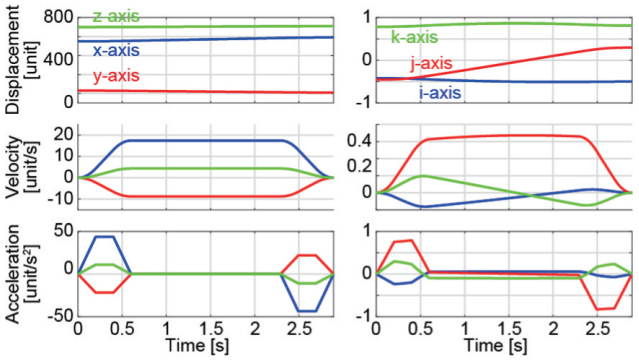
As shown in **Fig. 3**, the proposed decoupled approach for industrial robots can generate smooth joint trajectories to realize linear tool pose motion in the workspace.



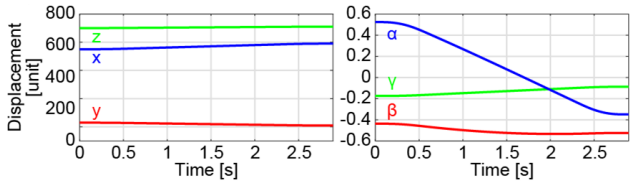
(a) Original commands and interpolated trajectory



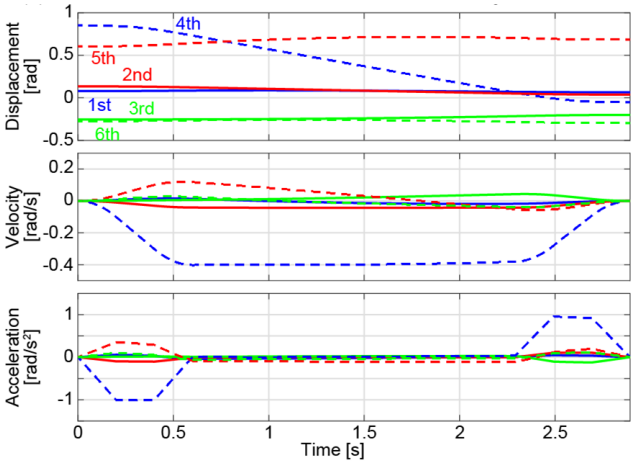
(b) Tangential velocity and angular velocity



(c) Kinematic profiles in workpiece coordinate system



(d) Kinematic profiles of pose vector



(e) Kinematic profiles in joint coordinate system

Fig. 3. Linear point-to-point tool-path interpolation using decoupled approach for industrial robots.

Table 1. Tool pose command vector for point-to-point tool-path interpolation.

Pose No.	Position [mm]			Orientation [deg]		
	x	y	z	α	β	γ
1	550	30	700	30	-25	-10
2	590	10	710	-20	-30	-5

3. Kinematic Local Corner Smoothing for Industrial Robots

For high-speed machining, it is necessary to generate smooth trajectories from the given commands. Usually, machining commands are given as a series of discrete points, such as G-codes. Therefore, the given command trajectory is continuous only in position and discontinuous in velocity and acceleration, and the tool motion is required to stop at the junction of each command to change the velocity direction. Otherwise, an infinite acceleration would be required to follow the command trajectory. Stopping the tool machine increases the control time and decreases productivity.

3.1. Kinematic Smoothing for Local Cornering Motion

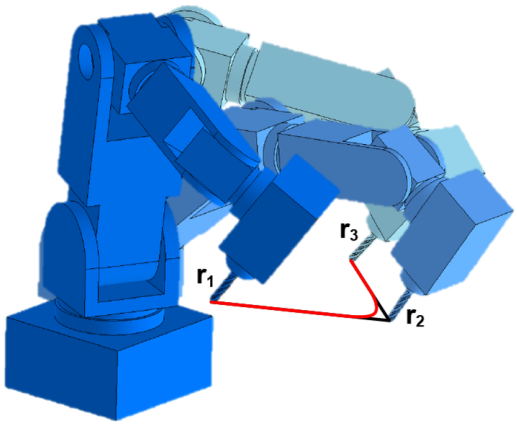
The principle behind the kinematic local corner smoothing algorithm is to generate a trajectory by directly smoothing the kinematic values in the joint coordinates instead of the workpiece coordinates. **Fig. 4** illustrates the proposed kinematic smoothing strategy for local corner smoothing. Smoothing the joint kinematics in the JCS leads to smooth tool motion in the WCS.

To generate a smooth kinematic profile at each joint, synchronized 3-segmented jerk profiles are used to generate each joint trajectory, as shown in **Fig. 4(c)**. The details of the 3-segmented jerk profiles are explained in Section 3.2.

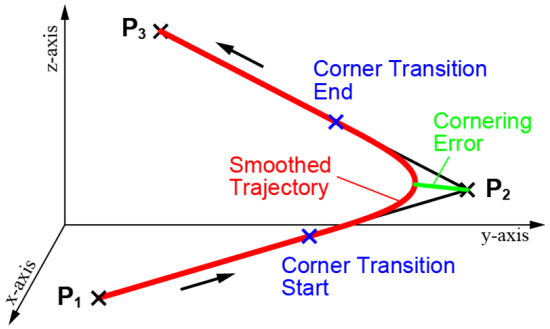
As shown in **Fig. 4(b)**, a cornering error appears when the trajectory is modulated for local corner smoothing. Therefore, the maximum deviation from the original corner must be controlled to keep the deviation within the user-specified cornering tolerance. In the proposed kinematic smoothing algorithm, this cornering error can be controlled by tuning the cornering duration of the kinematic profiles. The details of the error control strategy are described in Section 3.3.

3.2. 3-Segmented Jerk Profile for High-Speed Machining

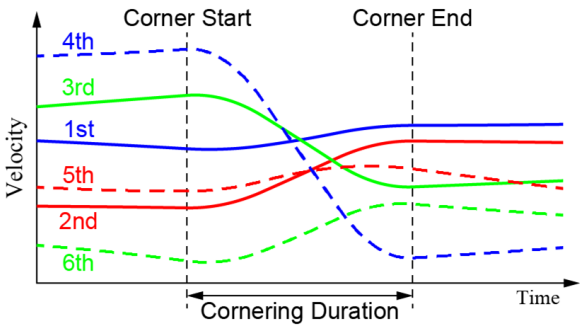
Local corner smoothing is effective for achieving high-speed machining. A 3-segmented jerk profile to generate a smooth kinematic profile between the start and end points of the corner is proposed here. **Fig. 5** illustrates the proposed 3-segmented jerk profile for kinematic local corner smoothing. This smoothing profile is composed of three jerk commands. The cornering duration is divided into three phases, each of which has a piecewise constant jerk value.



(a) Cornering motion of industrial robot



(b) Cornering trajectory in workspace



(c) Axis kinematic velocity profiles

Fig. 4. Kinematic local corner smoothing strategy for industrial robots.

To generate a trajectory with the proposed 3-segmented jerk profile, the displacement, velocity, and acceleration values at the start and end points of the cornering motion are required as kinematic boundary conditions. The boundary conditions for each axis are determined by the tool pose vector in the workspace. From the tool pose vectors at the start and end of the cornering motion, the position, velocity, and acceleration values for each axis are calculated using the inverse kinematics of the industrial robot. Based on the calculated kinematic values of each axis, smooth trajectories are generated using the proposed 3-segmented jerk profile.

As shown in **Fig. 5**, the acceleration increases from the initial value A_S with a constant jerk value during the first phase. This acceleration is also controlled during the second and third phases with a constant jerk value in each

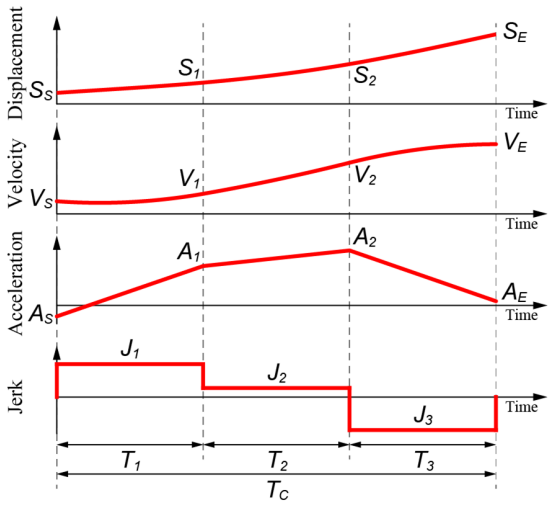


Fig. 5. Trajectory generation based on 3-segmented jerk profile.

phase until the end acceleration value A_E is reached.

The jerk profile for each phase of the cornering motion can be expressed as

$$j(t) = \begin{cases} J_1 & (0 \le t < T_1) \\ J_2 & (T_1 \le t < T_1 + T_2) \\ J_3 & (T_1 + T_2 \le t \le T_1 + T_2 + T_3) \end{cases}, \quad . \tag{8}$$

where J_1, J_2, and J_3 are the jerk values at each phase, and T_1, T_2, and T_3 are the time durations of each segment. The sum of the time durations equals the cornering duration: $T_1 + T_2 + T_3 = T_C$. The acceleration, velocity, and displacement profiles are calculated by integrating Eq. (8) with respect to time t. The displacement profile becomes

$$s(t) = \begin{cases} S_S + V_S t + \frac{A_S}{2} t^2 + \frac{J_1}{6} t^3 & (0 \le t < T_1) \\ S_1 + V_1 (t - T_1) + \frac{A_1}{2} (t - T_1)^2 & \\ \quad + \frac{J_2}{6} (t - T_1)^3 & (T_1 \le t < T_1 + T_2) \\ S_2 + V_2 (t - T_1 - T_2) + \frac{A_2}{2} (t - T_1 - T_2)^2 & \\ \quad + \frac{J_3}{6} (t - T_1 - T_2)^3 & (T_1 + T_2 \le t \le T_C) \end{cases} \tag{9}$$

where S_S, V_S, and A_S are the initial kinematic values of the cornering motion, and S_1, S_2, V_1, V_2, A_1, and A_2 are the kinematic values at the end of each phase. Here, the final displacement value can be obtained from the third equation by setting $t = T_1 + T_2 + T_3 = T_C$. This implies that the initial displacement value S_S has to become the final displacement value S_E at the end of the third phase. Similarly, the initial values of the velocity and acceleration, V_S and A_S, must reach the final values, V_E and A_E, at the end of the cornering profile.

These conditions provide kinematic constraints for determining the parameters of the cornering profile. To satisfy the kinematic boundary conditions imposed by the displacement, velocity, and acceleration at the start and end of the cornering motion, the kinematic constraints for the acceleration, velocity, and displacement are derived as

$$
\begin{cases}
\text{Acceleration constraint:} \\
\quad A_E = A_S + J_1T_1 + J_2T_2 + J_3T_3 \\
\text{Velocity constraint:} \\
\quad V_E = V_S + A_ST_1 + \dfrac{J_1}{2}T_1^2 \\
\qquad + (A_S + J_1T_1)\,T_2 + \dfrac{J_2}{2}T_2^2 \\
\qquad + (A_S + J_1T_1 + J_2T_2)\,T_3 + \dfrac{J_3}{2}T_3^2 \\
\text{Displacement constraint:} \\
\quad S_E = S_S + V_ST_1 + \dfrac{A_S}{2}T_1^2 + \dfrac{J_1}{6}T_1^3 \\
\qquad + \left(V_S + A_ST_1 + \dfrac{J_1}{2}T_1^2\right)T_2 + \dfrac{A_S + J_1T_1}{2}T_2^2 + \dfrac{J_2}{6}T_2^3 \\
\qquad + \left(V_S + A_ST_1 + \dfrac{J_1}{2}T_1^2 + (A_S + J_1T_1)\,T_2 + \dfrac{J_2}{2}T_2^2\right)T_3 \\
\qquad + \dfrac{A_S + J_1T_1 + J_2T_2}{2}T_3^2 + \dfrac{J_3}{6}T_3^3
\end{cases}
\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (10)
$$

Note that there are three constraints in the constraint equations in Eq. (10) for determining the six unknown parameters: the jerk values at each segment, J_1, J_2, and J_3, and the time duration of each phase, T_1, T_2, and T_3. To compute the parameters for the cornering motion, the durations of each segment are set to the same value of $T_1 = T_2 = T_3 = T_C/3$. Suitable jerk values for each segment can then be obtained to satisfy the kinematic constraints in Eq. (10).

By solving the three constraint equations in Eq. (10) analytically as a set of simultaneous equations, the constant jerk values for each segment are computed as

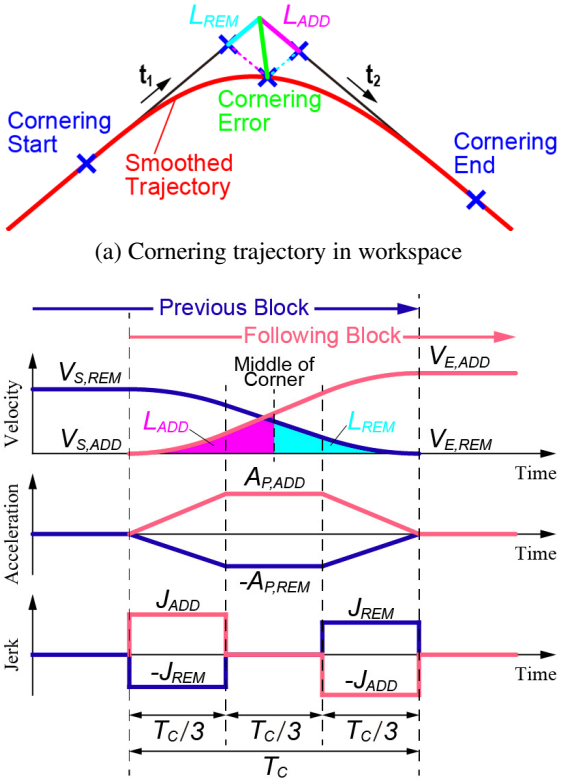
$$
\begin{aligned}
J_1 &= \frac{54\,(S_E - S_S) + 18\,(2V_S + V_E)\,T_C + (11A_S - 2A_E)\,T_C^2}{2T_C^3} \\
J_2 &= \frac{108\,(S_S - S_E) + 54\,(V_S + V_E)\,T_C + (7A_S - 7A_E)\,T_C^2}{2T_C^3} \\
J_3 &= \frac{54\,(S_E - S_S) + 18\,(V_S + 2V_E)\,T_C + (2A_S - 11A_E)\,T_C^2}{2T_C^3}
\end{aligned}
\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (11)
$$

where T_C is the total cornering time. The jerk values of the 3-segmented jerk profile can thereby be computed to generate a smooth local cornering trajectory. Here, the total cornering time needs to be determined to compute the jerk values, and this total cornering time will also affect the cornering error. The determination of the cornering time to control the cornering error for precision machining is described in the next section.

3.3. Error Control for Precision Machining

Controlling the cornering error caused by corner smoothing is important for high-precision machining. In the proposed kinematic local corner-smoothing algorithm for industrial robots, the cornering error can be controlled based on the time duration of the cornering trajectory.

As shown in **Fig. 6**, a cornering error occurs when the original sharp corner is replaced by a smooth cornering trajectory. The cornering error is defined at the middle of the cornering motion. The proposed kinematic smoothing algorithm generates a cornering trajectory by gradually changing the entry velocity into the corner to the exit velocity. This gradual change in the feed direction can be represented by the velocity of the block when it decelerates before the corner. This deceleration occurs over the same time duration as the acceleration of the block velocity after the corner. Here, the proposed kinematic smoothing method generates a velocity profile with a 3-segmented piecewise constant jerk. The represented velocity profile for calculating the cornering error at the middle of the cornering trajectory is therefore assumed to have a trapezoidal acceleration profile, as shown in **Fig. 6(b)**.



(a) Cornering trajectory in workspace

(b) Kinematic profiles of blocks

Fig. 6. Cornering error control based on trapezoidal acceleration profile.

The proposed trapezoidal acceleration profile for error estimation can be expressed using Eqs. (8) and (9) where the initial and final accelerations $A_S = A_E = 0$, and the jerk values of each segment are set as $J_1 = -J_3$ and $J_2 = 0$. If the time duration of each segment is set to identical values as $T_1 = T_2 = T_3 = T_C/3$, the peak acceleration value A_{Peak} will then be determined by the variation value ΔV of the initial and the final velocity, which is equal to the velocity of the block, as

$$
\Delta V = V_E - V_S = \frac{2}{3}A_{Peak}T_C \Rightarrow A_{Peak} = \frac{3\Delta V}{2T_C} \quad . \quad (12)
$$

The remaining and added distances can be calculated

from Eq. (9) using the intermediate time of the cornering motion $t = T_1 + T_2/2$ as

$$\begin{aligned} S_{mid} &= s\left(T_1 + \frac{T_2}{2}\right) \\ &= \frac{1}{6}\frac{A_{Peak}}{T_1}T_1^3 + \frac{1}{2}A_{Peak}T_1\frac{T_2}{2} + \frac{1}{2}A_{Peak}\left(\frac{T_2}{2}\right)^2 \\ &= \frac{13}{24}A_{Peak}\left(\frac{T_C}{3}\right)^2 \quad \ldots\ldots\ldots\ldots (13) \end{aligned}$$

Finally, by substituting A_{Peak} in Eq. (12) into the mid-distance S_{mid} in Eq. (13), the remaining and added distances, L_{REM} and L_{ADD}, can be calculated from the velocity variation ΔV (or F) and the cornering duration T_C as

$$L = \frac{13}{24}\left(\frac{3\Delta V}{2T_C}\right)\left(\frac{T_C}{3}\right)^2 = \frac{13\Delta V}{144}T_C = \frac{13F}{144}T_C \quad (14)$$

As shown in **Fig. 6**, the cornering error ε can be determined from the remaining distance of the block before the corner L_{REM} and the added distance of the block after the corner L_{ADD} as

$$\varepsilon = \|L_{ADD}\boldsymbol{t}_2 - L_{REM}\boldsymbol{t}_1\|, \quad \ldots\ldots\ldots\ldots (15)$$

where $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are the direction vectors of the block before and after the corner, respectively. Using the remaining and added distances from Eq. (14), the relationship between the cornering error and the cornering duration is expressed as

$$\begin{aligned} \varepsilon &= \left\| \frac{13F_2}{144}T_C\boldsymbol{t}_2 - \frac{13F_1}{144}T_C\boldsymbol{t}_1 \right\| \\ &= \frac{13}{144}\|F_2\boldsymbol{t}_2 - F_1\boldsymbol{t}_1\|\,T_C \quad \ldots\ldots\ldots\ldots (16) \end{aligned}$$

Therefore, the cornering duration T_C for the target cornering error ε can be expressed as

$$T_C = \frac{144}{13\|F_2\boldsymbol{t}_2 - F_1\boldsymbol{t}_1\|}\varepsilon \quad \ldots\ldots\ldots\ldots (17)$$

Kinematic local corner smoothing for industrial robots can then be realized by using this calculated cornering duration for the proposed 3-segmented jerk profile and considering the cornering tolerance.

3.4. Illustrative Examples

The application of the proposed kinematic local corner smoothing strategy to acute-angled and obtuse-angled corners is illustrated in this section.

Figure 7 shows the application of the proposed kinematic smoothing algorithm to smooth acute and obtuse corners. The pose vectors for both corners are listed in **Table 2**. As shown in **Fig. 7(a)**, the corners are defined by two linear translational motions on the XY plane with constant tool orientation commands. The cornering tolerance for both corners is set to 2 mm, and the feedrate is set to 100 mm/s.

As shown in **Figs. 7(a)** and **(b)**, the original commanded sharp corners are replaced by smooth cornering feed motion within the defined cornering tolerance. **Fig. 7(c)** shows the joint kinematic profiles of the cornering motions. The trajectories of each joint are generated using the smooth and continuous 3-segmented acceleration profiles.

As shown in **Figs. 7(b)** and **(c)**, the duration of the cornering motion is adjusted according to the corner angle to satisfy the cornering tolerance. When the corner angle is acute, the cornering duration becomes shorter to ensure adherence to the cornering tolerance. In comparison, when the corner angle is obtuse, the cornering duration

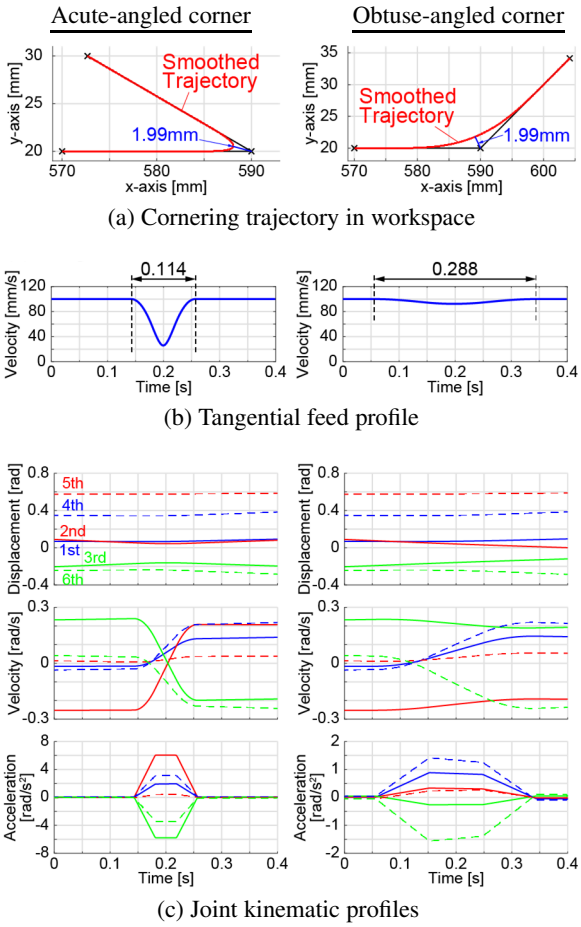


Fig. 7. Kinematic smoothing of acute-angled and obtuse-angled corner.

Table 2. Tool pose command vector for single corner tool-path interpolation.

(a) Acute-angled corner

Pose No.	Position [mm]			Orientation [deg]		
	x	y	z	α	β	γ
1	570.0	20.0	710.0	5	-25	-5
2	590.0	20.0	710.0	5	-25	-5
3	572.7	30.0	710.0	5	-25	-5

(b) Obtuse-angled corner

Pose No.	Position [mm]			Orientation [deg]		
	x	y	z	α	β	γ
1	570.0	20.0	710.0	5	-25	-5
2	590.0	20.0	710.0	5	-25	-5
3	604.1	34.1	710.0	5	-25	-5

becomes longer so that the amplitude of the acceleration profiles becomes moderate. As shown, the duration of the cornering motion is adjusted according to the corner angle to achieve precise robot machining.

These results demonstrate that the proposed kinematic local corner smoothing algorithm for industrial robots can generate smooth cornering motions according to the corner angle and cornering tolerance.

4. Simulation Result

The effectiveness of the kinematic smoothing algorithm proposed in the previous sections for a test tool path is demonstrated in this section.

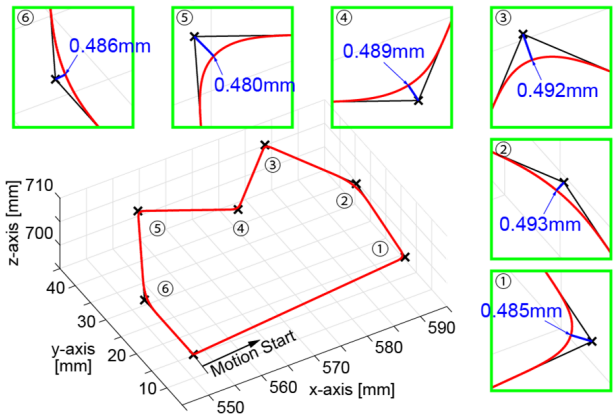
Figure 8 shows the tool-path used in the simulation and the interpolated trajectory. The tool-path consists of seven linear motion commands with eight command points for the tool position and orientation. The command pose vectors are presented in **Table 3**. The feedrate in each linear translational motion is set to 80 mm/s. For the original commands, which includes six corners, the proposed trajectory smoothing method is used to generate a smooth trajectory. The maximum tolerance between the original command and the generated trajectory is set to 0.5 mm.

As shown in **Figs. 8(a)** and **(b)**, the proposed algorithm generates a smooth motion trajectory at each corner for both TCP and ORI motions. The smooth motion trajectory is indicated by the tangential feedrate and angular velocity profiles shown in **Fig. 8(c)**, where the tool motion speed is decelerated in the cornering portion. The resulting errors at each corner are controlled by adjusting the cornering duration according to the corner angle. The trajectory at each corner of the ORI motion is smoothly interpolated in synchronization with the TCP motion.

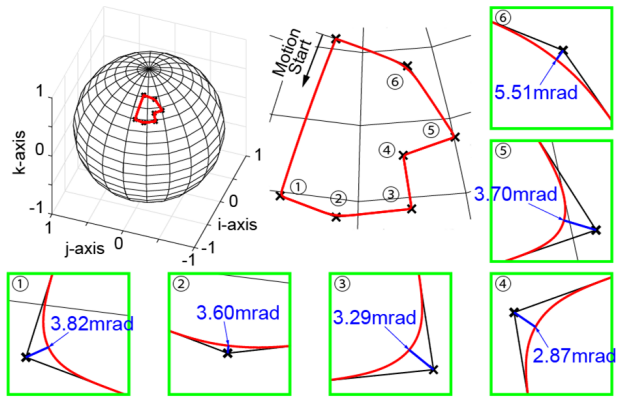
Figure 9 shows the kinematic profiles of the end-effector in Cartesian and spherical coordinates. As shown in **Fig. 9(a)**, a smooth and continuous trajectory is generated from the discrete command pose vectors given in **Table 3**. **Fig. 9(b)** shows the contouring error between the original tool-path and the smoothed trajectory generated by the proposed kinematic smoothing algorithm. As shown, the errors at each corner are controlled to be smaller than the cornering tolerance. In addition, the interpolated trajectory falls on the original tool-path in the straight sections, except at the corners. This means that there are no errors in the straight sections. These results demonstrate that the generated trajectory of the tool motion in the workspace is continuous and accurate.

Finally, **Fig. 10** presents the kinematic profiles of each joint in the JCS. As shown in **Fig. 10**, the proposed interpolation strategy smoothly generates the trajectory of each joint. Additionally, the acceleration profiles show that synchronized cornering joint motion is generated with the 3-segmented constant jerk profile.

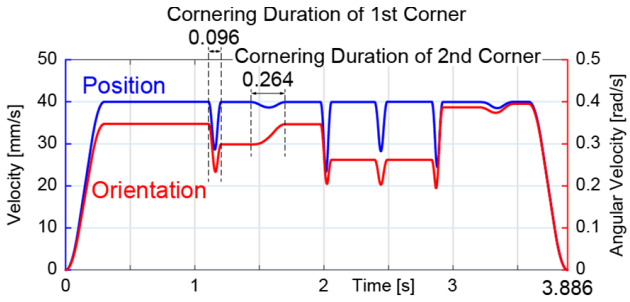
The results for the multi-segmented tool-path, show that the proposed kinematic smoothing algorithm can generate a smooth kinematic trajectory to achieve high-speed and high-accuracy robot machining.



(a) Smoothed trajectory of tool center point



(b) Smoothed trajectory of tool orientation vector

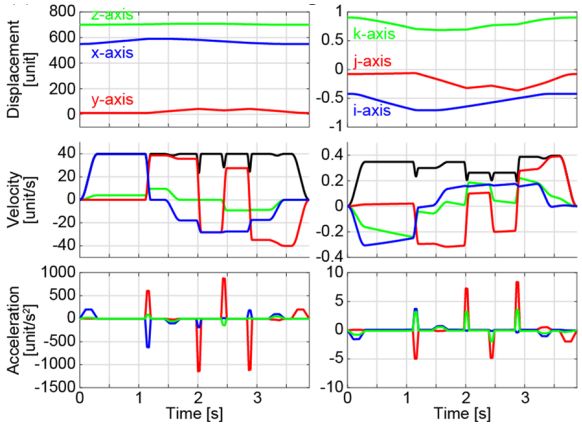


(c) Tangential feed-rate and angular velocity

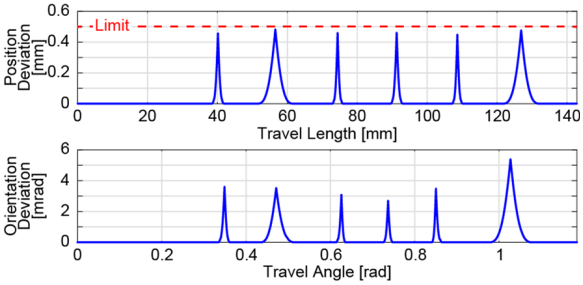
Fig. 8. Multi-segmented test tool-path and generated smooth trajectory.

Table 3. Tool pose command vector for multi-segmented tool-path.

Pose No.	Position [mm]			Orientation [deg]		
	x	y	z	α	β	γ
1	550.0	10.0	700.0	5	-25	10
2	590.0	10.0	704.0	5	-45	10
3	590.0	26.0	708.0	15	-45	15
4	582.0	42.0	708.0	25	-40	15
5	570.0	30.0	708.0	20	-35	15
6	558.0	42.0	704.0	25	-30	15
7	550.0	26.0	700.0	15	-25	15
8	550.0	10.0	700.0	5	-25	10



(a) Kinematic profiles in workpiece coordinates



(b) Contouring deviation of generated trajectory

Fig. 9. Kinematic profiles of pose vector along multi-segmented test tool-path.

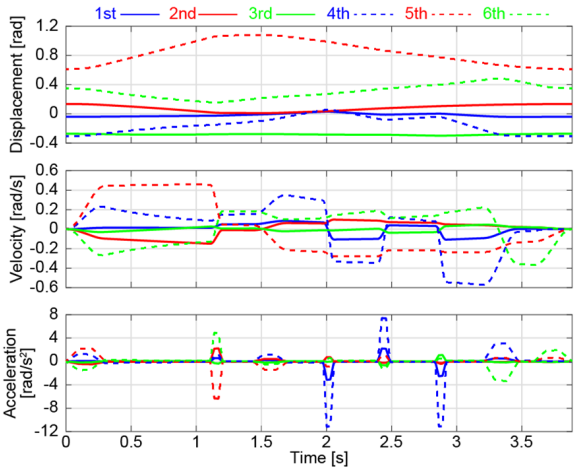


Fig. 10. Joint kinematic profiles generated by the proposed kinematic corner smoothing algorithm.

5. Conclusion

A novel trajectory generation algorithm to realize high-speed and high-precision machining using industrial robots was proposed in this paper.

The proposed decoupled approach for industrial robots interpolates discrete commands into continuous and smooth trajectories in workspace coordinates. In addition, the proposed kinematic local corner smoothing strategy generates smooth cornering motion using a 3-segmented piecewise jerk profile. The cornering error is considered by controlling the duration of the cornering motion.

The simulation results validate the effectiveness of the proposed kinematic smoothing strategy for generating smooth robot machining trajectories within the error tolerance.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP20K14622.

References:

[1] C. S. Chen and S. K. Chen, "Synchronization of tool tip trajectory and attitude based on the surface characteristics of workpiece for 6-DOF robot manipulator," Robotics and Computer-Integrated Manufacturing, Vol.59, pp. 13-27, 2019.

[2] Y. Chen and F. Dong, "Robot machining: recent development and future research issues," Int. J. of Advanced Manufacturing Technology, Vol.66, pp. 1489-1497, 2013.

[3] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, and L. T. Tunc, "Robots in machining," CIRP Annals, Vol.68, Issue 2, pp. 799-822, 2019.

[4] M. Beschi, S. Mutti, G. Nicola, M. Faroni, P. Magnoni, E. Villagrossi, and N. Pedrocchi, "Optimal robot motion planning of redundant robots in machining and additive manufacturing applications," Electronics, Vol.8, Issue 12, 1437, 2019.

[5] B. Greenway, "Robot accuracy," Ind. Robot, Vol.27, pp. 257-265, 2000.

[6] M. Dupac, "Smooth trajectory generation for rotating extensible manipulators," Mathematical Methods in the Applied Sciences, Vol.41, Issue 6, pp. 2281-2286, 2018.

[7] Y. Zhou, Z. Chen, and J. Tang, "A new method of designing the tooth surfaces of spiral bevel gears with ruled surface for their accurate five-axis flank milling," J. Manuf. Sci. Eng., Vol.139, No.6, 61004, 2017.

[8] F. Xiea, L. Chenb, Z. Lia, and K. Tang, "Path smoothing and feed rate planning for robotic curved layer additive manufacturing," Robotics and Computer Integrated Manufacturing, Vol.65, 101967, 2020.

[9] W. Wang, C. Hu, K. Zhou, S. He, and L. Zhu, "Local asymmetrical corner trajectory smoothing with bidirectional planning and adjusting algorithm for CNC machining," Robotics and Computer Integrated Manufacturing, Vol.68, 102058, 2021.

[10] L. Biagiotti and C. Melchiorri, "Online trajectory planning and filtering for robotic applications via B-spline smoothing filters," Proc. of 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 5668-5673, 2013.

[11] M. A. Funes-Lora, E. A. Portilla-Flores, E. Vega-Alvarado, R. Rivera Blas, E. A. Merchán Cruz, and M. F. Carbajal Romero, "A Novel Mesh Following Technique Based on a Non-Approximant Surface Reconstruction for Industrial Robotic Path Generation," IEEE Access, Vol.7, pp. 22807-22817, 2019.

[12] S. Tajima and B. Sencer, "Kinematic corner smoothing for high speed machine tools," Int. J. of Machine Tools and Manufacture, Vol.108, pp. 27-43, 2016.

[13] R. Béarée and A. Olabi, "Dissociated jerk-limited trajectory applied to time-varying vibration reduction," Robotics and Computer-Integrated Manufacturing, Vol.29, Issue 2, pp. 444-453, 2013.

[14] Y. Fang, J. Qi, J. Hu, W. Wang, and Y. Peng, "An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints," Mechanism and Machine Theory, Vol.153, 103957, 2020.

[15] S. Tajima and B. Sencer, "Accurate real-time interpolation of 5-axis tool-paths with local corner smoothing," Int. J. of Machine Tools and Manufacture, Vol.142, pp. 1-15, 2019.

[16] Y. Chen and B. Li, "A Piecewise Acceleration-Optimal and Smooth-Jerk Trajectory Planning Method for Robot Manipulator along a Predefined Path," Int. J. of Advanced Robotic Systems, Vol.8, Issue 4, pp. 184-193, 2011.

[17] Q. B. Xiao, M. Wan, Y. Liu, X. B. Qin, and W. H. Zhang, "Space corner smoothing of CNC machine tools through developing 3D general clothoid," Robotics and Computer-Integrated Manufacturing, Vol.64, 101949, 2020.

[18] W. Wang, C. Hu, K. Zhou, and S. He, "Corner trajectory smoothing with asymmetrical transition profile for CNC machine tools," Int. J. of Machine Tools and Manufacture, Vol.144, 103423, 2019.

[19] T. Haas, S. Weikert, and K. Wegener, "MPCC-based set point optimisation for machine tools," Int. J. Automation Technol., Vol.13, No.3, pp. 407-418, 2019.
[20] K. Nakamoto and Y. Takeuchi, "Recent advances in multiaxis control and multitasking machining," Int. J. Automation Technol., Vol.11, No.2, pp. 140-154, 2017.
[21] F. Sellmann, T. Haas, H. Nguyen, S. Weikert, and K. Wegener, "Geometry optimisation for 2D cutting: A quadratic programming approach," Int. J. Automation Technol., Vol.10, No.2, pp. 272-281, 2016.
[22] L. Zhang, K. Zhang, and Y. Yan, "Local corner smoothing transition algorithm based on double cubic NURBS for five-axis linear tool path," J. Mech. Eng., Vol.62, pp. 647-656, 2016.
[23] R. Sato, K. Morishita, I. Nishida, K. Shirase, M. Hasegawa, A. Saito, and T. Iwasaki, "Improvement of simultaneous 5-axis controlled machining accuracy by CL-data modification," Int. J. Automation Technol., Vol.13, No.5, pp. 583-592, 2019.
[24] F. Chen, J. Liao, J. Xiong, S. Yin, S. Huang, and Q. Tang, "High-precision trajectory tracking design and simulation for six degree of freedom robot based on improved active disturbance rejection control," J. of Mechanical Engineering Science, Vol.233, No.10, pp. 3659-3669, 2019.
[25] M. X. Kong, C. Ji, Z. S. Chen, and R. F. Li, "Smooth and near time-optimal trajectory planning of robotic manipulator with smooth constraint based on cubic B-spline," Proc. of 2013 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO), pp. 2328-2333, 2013.
[26] R. Zhao and S. Ratchev, "On-line trajectory planning with time-variant motion constraints for industrial robot manipulators," Proc. of 2017 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3748-3753, 2017.
[27] B. Boyacioglu and S. Ertugrul, "Time-optimal Smoothing of RRT-given Path for Manipulators," Proc. of the 13th Int. Conf. on Informatics in Control, Automation and Robotics, Vol.2, pp. 406-411, 2016.
[28] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, "Time-Optimal Path Tracking for Robots a Convex Optimization Approach," IEEE Trans. on Automatic Control, Vol.54, No.10, pp. 2318-2327, 2009.
[29] K. Hu, Y. Dong, and D. Wu, "Smooth time-optimal path tracking for robot manipulators with kinematic constraints," Proc. of ASME 2020 Int. Mechanical Engineering Congress and Exposition, Vol.7B, V07BT07A038, doi: 10.1115/IMECE2020-23637, 2020.
[30] J. Yang, D. Li, C. Ye, and H. Ding, "An-analytical C3 continuous tool path corner smoothing algorithm for 6R robot manipulator," Robotics and Computer Integrated Manufacturing, Vol.64, 101947, 2020.

Name:
Shingo Tajima

Affiliation:
Assistant Professor, Tokyo Institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan

Brief Biographical History:
2019 Received Dual Ph.D. from Oregon State University
2019- Assistant Professor, Tokyo Institute of Technology

Main Works:
• "Real-time trajectory generation for 5-axis machine tools with singularity avoidance," CIRP Annals, Vol.69, pp. 349-352, 2020.
• "Accurate real-time interpolation of 5-axis tool-paths with local corner smoothing," Int. J. of Machine Tools and Manufacture, Vol.142, pp. 1-15, 2019.
• "Accurate interpolation of machining tool-paths based on FIR filtering," Precision Engineering, Vol.52, pp. 332-344, 2018.

Membership in Academic Societies:
• Japan Society of Mechanical Engineers (JSME)
• Japan Society for Precision Engineering (JSPE)

Name:
Satoshi Iwamoto

Affiliation:
Master Course Student, Tokyo Institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan

Brief Biographical History:
2019 Received Bachelor's degree from Tokyo Institute of Technology
2020- Master Course Student, Tokyo Institute of Technology

Main Works:
• "Disturbance response characteristic of counterweight mechanism using reaction force," Proc. of JSPE Semestrial Meeting, pp. 103-104, 2020.

Name:
Hayato Yoshioka

Affiliation:
Associate Professor, Tokyo Institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan

Brief Biographical History:
2003- Assistant Professor, Tokyo Institute of Technology
2006- Associate Professor, Tokyo Institute of Technology

Main Works:
• "Design Concept and Structural Configuration of Advanced Nano-Pattern Generator with Large Work Area 'ANGEL'," Int. J. Automation Technol., Vol.5, No.1, pp. 38-44, 2011.
• "Micro patterning on curved surface with a fast tool servo system for micro milling process," CIRP Annals, Vol.69, No.1, pp. 325-328, 2020.

Membership in Academic Societies:
• Japan Society of Mechanical Engineers (JSME)
• Japan Society for Precision Engineering (JSPE)
• International Academy for Production Engineering (CIRP)
• European Society for Precision Engineering and Nanotechnology (euspen)