Lecture Notes in Mechanical Engineering

Salvatore Gerbino · Antonio Lanzotti ·
Massimo Martorelli ·
Ramón Mirálbes Buil · Caterina Rizzi ·
Lionel Roucoules *Editors*

Advances on Mechanics, Design Engineering and Manufacturing IV

Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2022, June 1–3, 2022, Ischia, Italy

Springer

Path Approximation Strategies for Robot Manufacturing: A Preliminary Experimental Evaluation

Matteo Bigliardi[1], Pietro Bilancia[1(✉)], Roberto Raffaeli[1], Margherita Peruzzini[2], Giovanni Berselli[3], and Marcello Pellicciari[1]

[1] DISMI, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy
239958@studenti.unimore.it, {pietro.bilancia,roberto.raffaeli,marcello.pellicciari}@unimore.it
[2] DIEF, University of Modena and Reggio Emilia, Via Vivarelli 10, 41125 Modena, Italy
margherita.peruzzini@unimore.it
[3] DIME, University of Genova, Via All'Opera Pia 15, 16145 Genova, Italy
giovanni.berselli@unige.it

Abstract. Industrial Robots (IRs) are increasingly adopted for material subtraction or deposition functions owing to their advantages over machine tools, like cost-effectiveness and versatility. Unfortunately, the development of efficient robot manufacturing processes still faces unsolved issues related to the IRs poor positioning accuracy and to the tool path generation process. Novel engineering methods and tools are needed for CAD based programming of accurate paths and continuous robot motions to obtain the required manufacturing quality and tolerances. Within this context, to achieve smoothness along the tool path formed by linear G-code segments, the IR controllers' approximation strategies, summarily reported in the manufacturer's manuals, must be considered. The aim of this paper is to present the preliminary work carried out to identify the approximation algorithms of a Kuka IR when executing linear moves. An experimental study is conducted by varying the controller settings and the maximum translational velocity. The robot behavior has been acquired thanks to the controller tracing function and then processed to yield relations readily employable for the interpretation of G-Code commands and the subsequent generation of proper robot motion instructions. The obtained formulas allow to accurately predict the robot geometric path and kinematics within the corner transition between two linear segments.

Keywords: Manufacturing robots · Path approximation · Corner smoothing · Robot programming · G-code translation

S. Gerbino et al. (Eds.): JCM 2022, LNME, pp. 380–389, 2023.
https://doi.org/10.1007/978-3-031-15928-2_33

1 Introduction

With the advent of Industry 4.0, manufacturing companies are geared towards an intelligent, more efficient and customized production to be achieved at reasonable costs [1]. In this context, Industrial Robots (IRs) have gradually been introduced in modern production systems as they enable the development of fully automated reconfigurable plants [2]. Traditional IRs tasks include assembly operations, part handling, picking and placing, products packing and inspection, where no strict tolerances in terms of position accuracy are demanded, except for a limited subset of points, normally teached and further checked during the calibration process [3].

As the improvement of their dynamic performance, IRs use has been extended also to industrial machining operations, such as milling, deburring, polishing and grinding [4–6]. In parallel, robotic-assisted additive manufacturing solutions are getting attention from many industrial sectors as they allow to fabricate medium-to-large scale customized components with reduced amount of supporting structures [7]. During the execution of these tasks, the robot contouring performance along the whole tool path becomes an aspect of primary importance since it strictly determines the geometric quality and tolerances of the manufactured parts. According to [4], this is mostly governed by the robot dynamic position and motion accuracy in the working area and by the tool path planning. The former comprises various inherent problems of serial manipulators, including link geometrical errors [8], joint backlash [6,9,10] and compliance [11], whose effects need to be carefully assessed and compensated through dedicated measuring systems (e.g. a laser tracker, as in [12,13]).

Concerning the path planning, although the capabilities of robot controllers have been steadily increased over the last years with the introduction of higher order (e.g. spline) motions, robot machining programs are usually generated from CAD models using CAM software in G-code standards [14,15]. In fact, the original curvilinear contour of the designed part is typically approximated through a series of linear segments between discrete points (G01 motion commands, as visible in Fig. 1). The resulting discretized path, consisting of straight lines and sharp turns, is continuous only in position, while velocity discontinuities occur due to differently oriented tangent vectors in the connecting segments [16]. Consequently, the tool motion is necessarily stopped at each corner so as to change the velocity direction. Frequent variations of the feed rate inevitably affect the manufacturing quality in terms of increased overall process time and poor geometric tolerances and surface quality [17]. With the aim to overcome these issues, plenty of corner smoothing algorithms have been proposed and validated in the recent academic literature (see [18] for a comparison).

In the same way, most of the commercial robot controllers integrate smoothing functions to achieve continuous motions between consecutive linear/circular blocks. These become effective in the control system once certain parameters are defined in the robot program. However, the rather limited availability of detailed documentations largely justifies the scarce utilization of such smoothing functions for the offline programming of robotic manufacturing cells. Owing to these

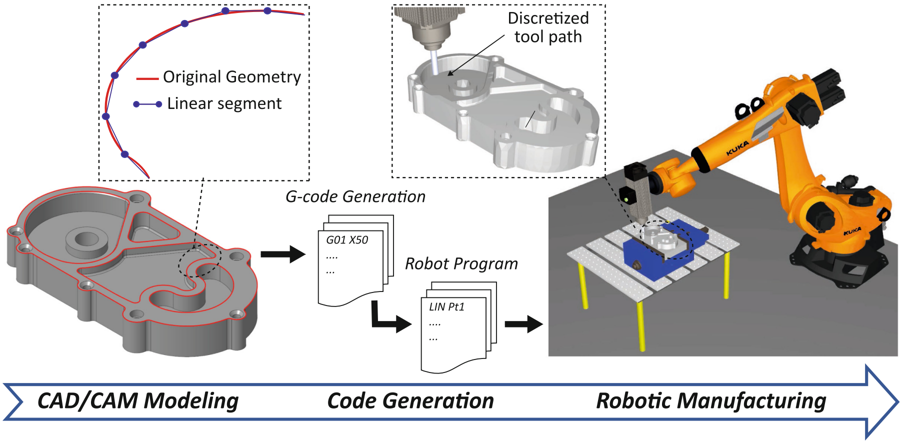

Fig. 1. Standard procedure for robot path generation in manufacturing tasks.

considerations and with the aim to provide practical design considerations and resources for the effective conversion of G-codes into robot programs for precise manufacturing tasks, the present paper provides an overview of the engineering methods for path approximation strategies of modern industrial controllers. The research focuses on the Kuka KRC4 model, although similar features are seen with different industrial controllers such as ABB and Stäubli. In particular, after a detailed description of the corner approximation curve, the experimental study conducted with a Kuka KR210 R2700 Prime robot is reported and the influence of the main geometrical and kinematic parameters on the approximated path is discussed.

The rest of paper is organized as follows: Sect. 2 recalls the standard path planning procedures for robot manufacturing; Sect. 3 presents a simplified analytical model of the Kuka corner smoothing function; Sect. 4 reports the experimental validation performed with a Kuka IR; Sect. 5 provides the concluding remarks.

2 Path Planning in Robot Manufacturing

For the purpose of robot path planning for manufacturing, the part geometry is first processed within dedicated CAD/CAM environments to generate the location file containing a set of discrete points in the Cartesian space, as shown in Fig. 1. From the IR controller side, these are reference points the manipulator has to pass through. While robot programs characterized by thousands of straight-line motion instructions are quite simple to conceive, the lack of higher order continuity (C^n, with $n \geq 1$) at corners strongly impacts the process since constant feed motion cannot be achieved. The resulting end-effector's movements, characterized by numerous accelerated/decelerated profiles, could also induce the excitation of robot structural modes, which would further deteriorate the overall

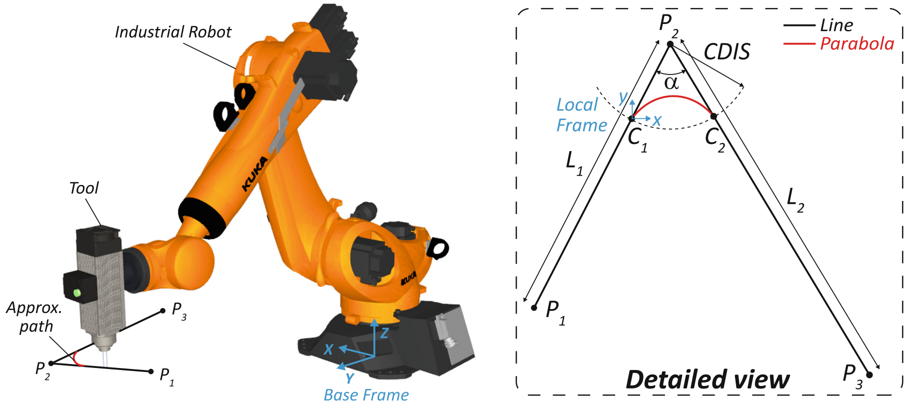

Fig. 2. Parametric schematic of the robot corner transition path.

position accuracy. To smooth the transition from one linear segment to the next, sharp corners can be replaced with properly defined parametric curves. Several corner smoothing methods have been considered in literature which deliver C^2 [14,19] or even C^3 [18,20] transition functions, i.e. respectively acceleration or jerk continuous feed motions. Despite their proven effectiveness, such functions result computationally expensive for real-time implementation, that is why are not integrated in the standard motion planning algorithms of commercial IR controllers. Their operability requires optional control packages such as the *Kuka Robot Sensor Interface* or the *ABB Externally-guided motion* (normally supplied upon extra costs), which enable the robot to realize generalized Cartesian paths generated from an external computer [13].

To simplify the overall control system and avoid costly software solutions, the computationally efficient C^1 profiles provided by IR controllers can be adopted. Following this approach, in the next section the smoothing method utilized by the Kuka KRC4 controller is analyzed and simple analytical relations are derived which can be utilized for optimizing the robot path planning and offline programming. It should be once again recalled that most of the existent G-code conversion tools still do not consider the transition functions available within the IR controller as no clear explanations regarding the corner blending algorithms are provided in the vendor manuals.

3 C^1 Corner Smoothing Approach

The transition between two adjacent lines defined by points P_1, P_2 and P_3 in the robot base coordinate system, i.e. $\overline{P_1P_2}$ and $\overline{P_2P_3}$ with length L_1 and L_2 respectively and intersection angle α, is a parabolic profile calculated within the Kuka controller for the approximate positioning contour. According to [21], if the end-effector translational velocity and acceleration are set identical for both the linear blocks through the parameters $\$VEL.CP$ and $\$ACC.CP$ (in the following referred to as $v_{L,max}$ and $a_{L,max}$ respectively), the parabola will be symmetrical about the bisector of α. The positions C_1 and C_2 at which the parabola joins the linear blocks are defined by the distance parameter $\$APO.CDIS$ (in the following named $CDIS$), as shown in Fig. 2.

With reference to the same figure, the parabola geometric equation can be determined analytically. In particular, starting from the generic second order polynomial function:

$$y = ax^2 + bx \tag{1}$$

by placing the reference frame in C_1 and then imposing the tangency condition at points C_1 and C_2 with the first and second linear segments ($\overline{P_1P_2}$ and $\overline{P_2P_3}$) respectively, i.e.:

$$\begin{cases} y'_{parab}(x) = y'_{line}(x), x = x_{C_1} \\ y'_{parab}(x) = y'_{line}(x), x = x_{C_2} \end{cases} \rightarrow \begin{cases} 2ax_{C_1} + b = \frac{\cos(\alpha/2)}{\sin(\alpha/2)} \\ 2ax_{C_2} + b = -\frac{\cos(\alpha/2)}{\sin(\alpha/2)} \end{cases} \tag{2}$$

where $x_{C_1} = 0$ and $x_{C_2} = 2CDIS\sin(\alpha/2)$, one could find:

$$a = -\frac{\cos(\alpha/2)}{2CDIS\sin^2(\alpha/2)} \quad b = \frac{\cos(\alpha/2)}{\sin(\alpha/2)} \tag{3}$$

From a kinematic viewpoint, by adopting the same reference frame (see Fig. 2), the parabolic motion can be seen as a combination of two contributions, i.e. a constant law and an accelerated law along the x- and y-direction respectively:

$$\begin{cases} x = v_x t \\ y = v_y t - 0.5a_P t^2 \end{cases} \tag{4}$$

where $v_x = v_{L,max}\sin(\alpha/2)$, $v_y = v_{L,max}\cos(\alpha/2)$ and a_P is the acceleration utilized during the parabolic motion along the y-direction. Solving the system expressed in Eq. 4 gives:

$$y = -\frac{a_P}{2v_{L,max}^2\sin^2(\alpha/2)}x^2 + \frac{\cos(\alpha/2)}{\sin(\alpha/2)}x \tag{5}$$

Hence, by comparing the geometric a and b (Eq. 3) with the corresponding terms in Eq. 5, the following expression for a_P can be derived:

$$a_P = v_{L,max}^2\frac{\cos(\alpha/2)}{CDIS} \tag{6}$$

From the above relations it emerges that the robot behavior when traveling the approximated corner path is influenced by a limited set of input parameters. This simple analytical formulation of the line-to-line smoothed transition does not require iterative calculations and offers design insight for the interpretation of G-Code files and the subsequent optimal generation of robot programs for manufacturing.

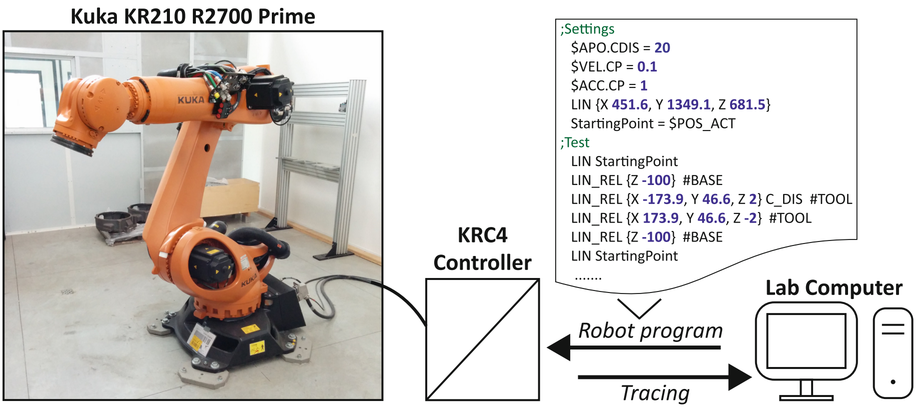

Fig. 3. Experimental platform.

4 Experiments

The reported path approximation method is verified experimentally on a Kuka KR210 R2700 Prime robot, as shown in Fig. 3. The utilized IR has an overall mass of 1111 kg, a maximum reach of 2696 mm and a maximum payload of 210 kg. It is mounted on the floor and connected to its cabinet, a Kuka KRC4 featuring the software KSS version 8.3.25 [22]. The tests have been performed by enforcing the IR end-effector through two consecutive straight lines (LIN commands) programmed in the operational space. A simple script is created which contains the main parameters for the corner transition and the related motion instructions (see Fig. 3). For each experiment, the end-effector Cartesian path is exported via the Kuka tracing function by monitoring the commanded position (channels X_CmdIpo, Y_CmdIpo and Z_CmdIpo) every 12 ms. To study the influence of $CDIS$, $v_{L,max}$ and $a_{L,max}$ on the line-to-line path approximation, two parametric studies have been performed. The numerical values of each parameter are listed in Table 1.

Table 1. Parameters tested in the experiments.

Parameter	Unit	First study	Second study
$L_1 = L_2$	m	0.23	0.23
α	deg	75	75
$CDIS$	m	$\{0.02, 0.06, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2\}$	0.08
$v_{L,max}$	m/s	$\{0.2, 0.4, 0.6\}$	$\{0.2, 0.4, 0.6\}$
$a_{L,max}$	m/s^2	4	$\{2, 2.5, 3, 3.5, 4\}$

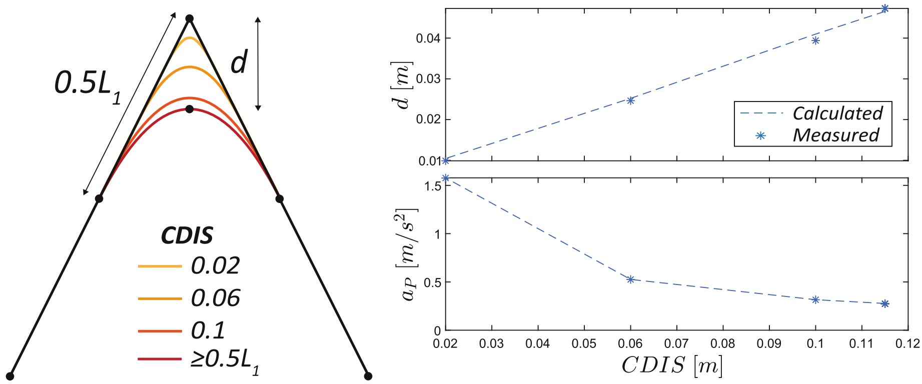

Fig. 4. Effect of varying $CDIS$ on the transition parabola.

At first, results involving $CDIS$ are presented in Fig. 4. On the left side it can be observed that the maximum accepted value for $CDIS$ is $L_1/2$, while upper values will automatically be truncated by the controller. Then, the right plot compares the cornering error d and acceleration a_P measured during the experiments with the ones calculated through Eqs. 3 and 6 (where the effective $CDIS$ values are considered). In particular, the theoretical value of d is calculated as the distance along the y-axis between P_2 and the parabola at $x_{P_2} = CDIS\sin(\alpha/2)$, whereas a_P is simply obtained by substituting the utilized $v_{L,max}$, α and $CDIS$ within Eq. 6. Overall, the curves are in good agreement, being the differences possibly due to the discretization errors during the sampling. The second group of experiments confirmed that the geometric transition curve (described from Eq. 1 with coefficients a and b as in Eq. 3) is not affected by $v_{L,max}$ and $a_{L,max}$ as long as the motors limits are not violated. Also, from the results shown in Fig. 5, where the translational velocity, acceleration and jerk (absolute values) are reported for each case, the following considerations can be made:

- the imposed $v_{L,max}$ (i.e. 0.2 m/s, 0.4 m/s and 0.6 m/s respectively) is always reached by the robot when traveling the linear segments (see the green circles);
- the value of $a_{L,max}$ set in the robot program is effective only when $v_{L,max}/a_{L,max} > k$, with $k = 0.168$, otherwise it will be automatically changed to $v_{L,max}/k$ from the controller;
- the maximum jerk can be estimated as $a_{L,max}/k$ where $a_{L,max}$ is the originally value set by the user;
- the C^1 cornering works for every tested parameter set and the robot motion has not to be stopped in the transition zone;
- the values of a_P are consistent with the ones calculated with Eq. 6 (0.39 m/s^2, 1.58 m/s^2 and 3.55 m/s^2 when $v_{L,max}$ equals 0.2 m/s, 0.4 m/s and 0.6 m/s).

The acquired data demonstrated the validity of the discussed analytical model and provided important information for the prediction of the robot behavior in the line-to-line transition. The accurate knowledge of the controller logic

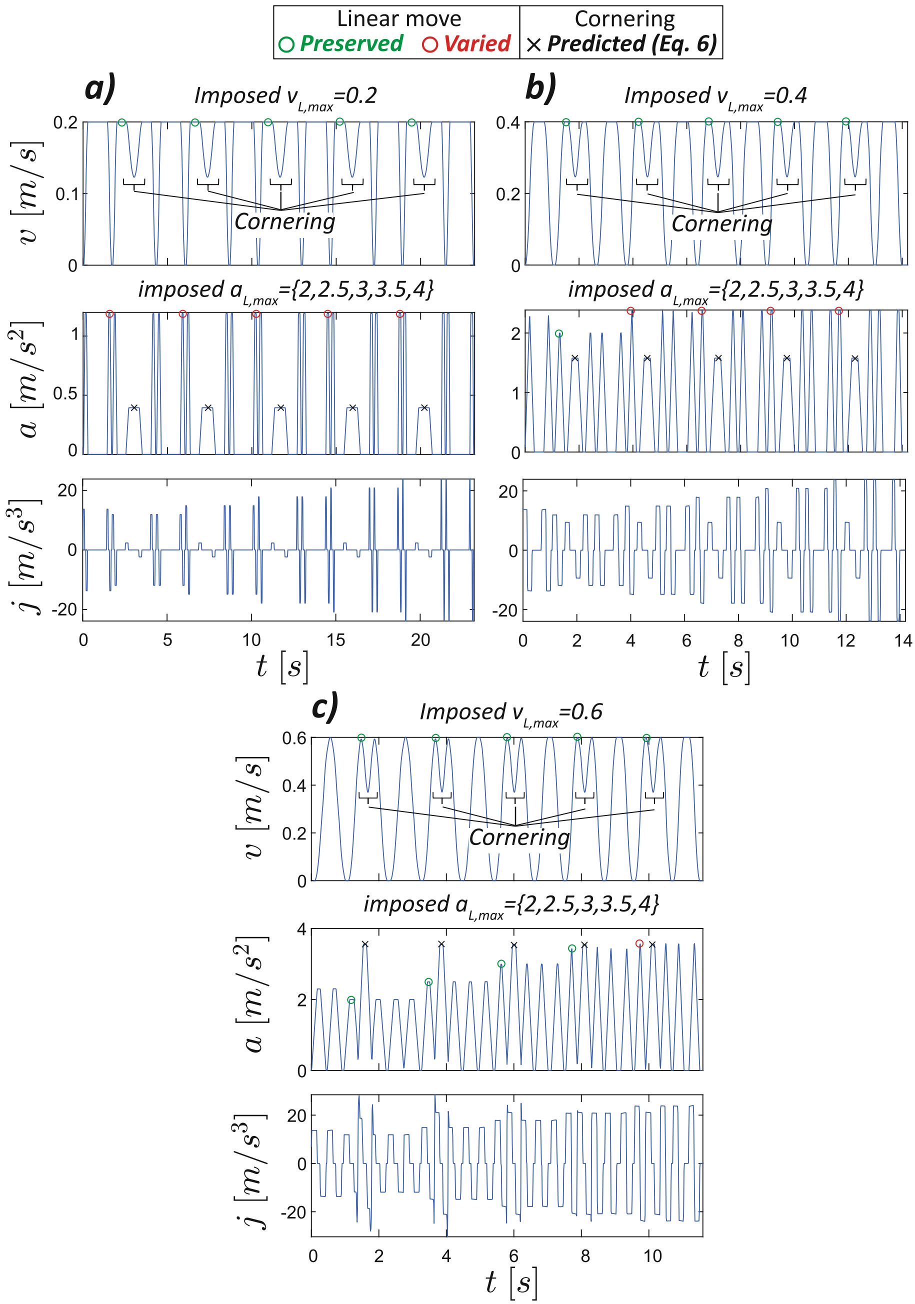

Fig. 5. Reference end-effector's velocity, acceleration and jerk profiles calculated within the KRC4 controller: experiments with a) $v_{L,max} = 0.2\,\text{m/s}$, b) $v_{L,max} = 0.4\,\text{m/s}$ and c) $v_{L,max} = 0.6\,\text{m/s}$.

can be integrated within a virtual prototype for the computer aided robotic process in which feed motion planning algorithms efficiently schedule the robot motion along corner-blended tool paths. As a limit, in case of specific applications demanding constant traveling speed, the investigated C^1 transitions can no longer be considered.

5 Conclusions

The paper discussed the path approximation strategies for robot manufacturing focusing on the smoothing algorithms utilized within the Kuka KRC4 controller when approaching consecutive linear segments. After a preliminary discussion of the criteria and requirements normally adopted from standard IRs for the calculation of the approximated paths, simplified analytical formulas of the C^1 parabolic transition curve have been evaluated. In the second part of the paper, an experimental campaign has been carried out on a Kuka IR by considering several combinations of geometric and kinematic parameters. The robot is programmed to execute linear paths with different approximation settings and its behavior is monitored thanks to the integrated tracing function. Within the explored design space, the simplified model and the experiments are found to be in good agreement and the transition curve can be easily predicted. In particular, both the cornering error and the acceleration module are captured from the model. Also, important considerations are made regarding the robot kinematic behavior when traveling the linear segments. Empirical relations are found which enable to predict the controller adjustments with respect to the imposed acceleration values. The presented model and the achieved experimental results will be of primary importance for future developments, in particular they will be embedded in virtual prototypes with whom build digital twins of robotic manufacturing cells and first-time-right CAD to Part engineering tools and methods.

Acknowledgments. This research was funded by the European Community's HORIZON 2020 programme under grant agreement No. 958303 (PENELOPE).

References

1. Vaidya, S., Ambad, P., Bhosle, S.: Industry 4.0-a glimpse. Proc. Manuf. **20**, 233–238 (2018)
2. Oztemel, E., Gursev, S.: Literature review of industry 4.0 and related technologies. J. Intell. Manuf. **31**(1), 127–182 (2020)
3. Nubiola, A., Bonev, I.A.: Absolute calibration of an ABB IRB 1600 robot using a laser tracker. Robot. Comput. Integrat. Manuf. **29**(1), 236–245 (2013)
4. Verl, A., Valente, A., Melkote, S., Brecher, C., Ozturk, E., Tunc, L.T.: Robots in machining. CIRP Ann. **68**(2), 799–822 (2019)
5. Chen, Y., Dong, F.: Robot machining: recent development and future research issues. Int. J. Adv. Manuf. Technol. **66**(9), 1489–1497 (2013)
6. Lehmann, C., Pellicciari, M., Drust, M., Gunnink, J.W.: Machining with industrial robots: the comet project approach. International Workshop on Robotics in Smart Manufacturing, pp. 27–36 (2013)

7. Urhal, P., Weightman, A., Diver, C., Bartolo, P.: Robot assisted additive manufacturing: a review. Robot. Comput. Integrat. Manuf. **59**, 335–345 (2019)
8. Lattanzi, L., Cristalli, C., Massa, D., Boria, S., Lépine, P., Pellicciari, M.: Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task. Int. J. Adv. Manuf. Technol. **111**(7), 1813–1829 (2020) Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task.The International Journal of Advanced Manufacturing Technology **111**(7), 1813–1829 (2020)
9. Slamani, M., Nubiola, A., Bonev, I.A.: Modeling and assessment of the backlash error of an industrial robot. Robotica **30**(7), 1167–1175 (2012)
10. Belloni, M., Bilancia, P., Raffaeli, R., Peruzzini, M., Pellicciari, M.: Design of a test rig for tuning and optimization of high dynamics servo-mechanisms employed in manufacturing automation. Proc. Manuf. **55**, 48–55 (2021)
11. Slamani, M., Bonev, I.A.: Characterization and experimental evaluation of gear transmission errors in an industrial robot. Indust. Robot Int. J. (2013)
12. Shi, X., Zhang, F., Qu, X., Liu, B.: An online real-time path compensation system for industrial robots based on laser tracker. Int. J. Adv. Robot. Syst. **13**(5), 1729881416663366 (2016)
13. Wang, Z., Zhang, R., Keogh, P.: Real-time laser tracker compensation of robotic drilling and machining. J. Manuf. Mater. Process. **4**(3), 79 (2020)
14. Li, G., Liu, H., Liu, S., Xiao, J.: A general c2 continuous toolpath corner smoothing method for a 5-DOF hybrid robot. Mech. Mach. Theory **169**, 104640 (2022)
15. Zeng, G., Chen, C.Y., Huang, D., Zhu, Y.: Robotic trajectory planning based on cl data. In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1723–1728 (2015)
16. Tajima, S., Iwamoto, S., Yoshioka, H.: Kinematic tool-path smoothing for 6-axis industrial machining robots. Int. J. Autom. Technol. **15**(5), 621–630 (2021)
17. Yang, J., Yuen, A.: An analytical local corner smoothing algorithm for five-axis CNC machining. Int. J. Mach. Tools Manuf. **123**, 22–35 (2017)
18. Yang, J., Li, D., Ye, C., Ding, H.: An analytical c3 continuous tool path corner smoothing algorithm for 6r robot manipulator. Robot. Comput. Integrat. Manuf. **64**, 101947 (2020)
19. Sencer, B., Ishizaki, K., Shamoto, E.: A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. Int. J. Adv. Manuf. Technol. **76**(9), 1977–1992 (2015)
20. Peng, J., Huang, P., Ding, Y., Ding, H.: An analytical method for decoupled local smoothing of linear paths in industrial robots. Robot. Comput. Integrat. Manuf. **72**, 102193 (2021)
21. Kuka, R.: Operating and programming instructions for system integrators (KSS 8.3 SI v7 EN ed.). KUKA Roboter GmbH, Germany (2018)
22. Gadaleta, M., Berselli, G., Pellicciari, M., Grassia, F.: Extensive experimental investigation for the optimization of the energy consumption of a high payload industrial robot with open research dataset. Robot. Comput. Integrat. Manuf. **68**, 102046 (2021)