machines

MDPI

Article

Quantitative Evaluation of an Industrial Robot Tool Trajectory Deviation Using a High-Speed Camera

Mantas Makulavičius, Sigitas Petkevičius, Vytautas Bučinskas and Andrius Dzedzickis *

Department of Mechatronics, Robotics and Digital Manufacturing, Vilnius Gediminas Technical University, LT-10105 Vilnius, Lithuania; m.makulavicius@vilniustech.lt (M.M.); sigitas.petkevicius@vilniustech.lt (S.P.); vytautas.bucinskas@vilniustech.lt (V.B.)

* Correspondence: andrius.dzedzickis@vilniustech.lt

Abstract

One of the primary applications of industrial robots is in various manufacturing processes, such as milling, grinding, and additive manufacturing. To achieve the desired precision in tool trajectory performance when machining specific parts, it is necessary to calibrate the tool centre point (TCP) of the robot for each manufacturing process. The development of industrial robot tool trajectories is a multipurpose task. It encompasses issues related to robot geometry, path interpolation type, and trajectory waypoints approximation. The primary objective of this study is to establish a camera-based methodology for evaluating trajectory-following accuracy in industrial robots. The present paper proposes the use of a high-speed motion camera system for non-contact tracking of TCP trajectories. By capturing the robot's end-effector motion in real-time and under actual trajectory tracking conditions, this technique enables a clearer understanding of how trajectory execution accuracy varies with velocity, trajectory geometry, trajectory interpolation, and robot kinematics. Provided analysis of two industrial robot types opened interesting findings related to the dependencies between the implementation of first- and second-degree interpolations. To illustrate this point, the implementation of second-degree interpolation ensures a more consistent velocity in the trajectory. This contrasts with first-degree interpolation, which is more challenging to achieve and is susceptible to variations in curvature. Conversely, the utilization of first-degree interpolation facilitates enhanced performance accuracy for smaller curvatures. The results of the experimental research confirm the initial hypothesis regarding the influence of interpolation mode and pave the way for future uses of this information for machine learning algorithms.

Keywords: industrial robotics; trajectory accuracy; toolpath generation; high-speed camera; non-contact measurements

Check for updates

Academic Editor: Mingyu Ding

Received: 3 November 2025
Revised: 15 December 2025
Accepted: 18 December 2025
Published: 19 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In modern manufacturing, industrial robots are increasingly being used for high-accuracy machining tasks that were traditionally dominated by Computer Numerical Control (CNC) machines. Applications include robotic milling or deburring, and additive manufacturing, like 3D printing, uniform material deposition, or distribution. It demands not only mechanical flexibility but also a high degree of machining path (or trajectory) accuracy to ensure quality, consistency, and productivity [1]. A critical aspect of performance in these applications is the robot's ability to accurately follow the programmed tool centre point (TCP) trajectory during complex, high-speed motions. As modern control

systems continue to evolve and improve, the demand for higher positioning accuracy has also increased [2]. However, attaining perfect motion accuracy is challenging due to various sources of error. Inherent limitations in robotic structures, including joint compliance (geometric errors), backlash, dynamic response delays (non-geometric errors), and interpolation inaccuracies, often result in deviations from the intended tool paths, particularly at elevated speeds and during curved or segmented trajectories [3]. Some research also raises the consideration that the open kinematic linkage of industrial robots reduces structural rigidity, which in turn affects machining accuracy [4]. In the field of robotic material distribution, the need for velocity and accuracy constraints is applied, when even distribution prevents localized stresses that can compromise material integrity, as seen in the previous literature [5]. Here, the balance between speed and trajectory accuracy control is essential for effective material distribution, ensuring uniform layer thickness and properties across all plates during the robotic process.

It is imperative to explicitly indicate the evaluation of the impact of linear interpolation, which involves the discretization of trajectories into linear segments. It introduces approximation errors, particularly in curved or complex trajectories. Furthermore, the software and industrial robot controllers enable the programming and execution of specialized trajectory interpolation techniques, such as generating a circular movement shape. This is achieved by defining three key points: the start point, the intermediate point, and the final point of the arc. Consequently, this technique enables a notable reduction in the program code length. However, it is a time-consuming process, as the key points must be selected manually [6]. These deviations, though sometimes small in magnitude, can accumulate and significantly affect surface quality, dimensional accuracy, and overall process reliability. Consequently, quantitative assessments of TCP trajectory-following accuracy are mandatory for two reasons. First, to ensure quality assurance. Second, to validate and optimize robot programming and control strategies [7].

Industrial robots frequently utilize various motion planning algorithms, wherein reduced interpolation quality leads to less precise motion trajectories, increased trajectory deviations, and augmented positioning errors [8]. A review of studies in robotic machining has identified trajectory interpolation approximation, or chord error, as a significant factor contributing to trajectory inaccuracy [9]. Consequently, inadequate interpolation quality, characterized by substantial chord errors, directly impairs motion accuracy and elevates contouring error in tasks necessitating trajectory following. A multitude of other works explored the issue of trajectory errors through improvements to interpolation techniques. For instance, researchers explored the use of adaptive interpolation, which adjusts the step size based on curvature and speed to maintain error bounds. Additionally, real-time error compensation has been investigated, wherein external sensors (e.g., laser trackers or cameras) correct the trajectory in real-time [10,11]. The following paper [10] proposes a real-time method for detecting trajectory errors, suitable for industrial robots. This method addresses the challenges posed by continuous trajectory deviations and unknown, time-varying speeds. The system utilizes Continuous Dynamic Time Warping (CDTW) for real-time estimation of position error, incorporating laser tracking technology. A multimodal fusion-based monitoring framework for the detection of robot trajectory deviations during wire arc additive manufacturing is presented in another research [12]. The system integrates visual sensing through a monochrome Charge-Coupled Device (CCD) camera and thermal sensing via an infrared imager. This integration enables the capture of both geometric and temperature characteristics of the weld environment, and implements neural networks, allowing control of the deviation. There are some novel photogrammetry-based methods for industrial robots controlled by 3D camera sensors [13]. Unlike modern methods, the proposed method allows simultaneous calibration of the robot kinematics with the attached

camera on the robotic arm. Another research example [1] proposes the development of a digital model of machining, providing a tool for virtual process commissioning that can be employed during the process planning stage and additionally evaluating robot stiffness based on workpiece placement.

While laser tracking systems, Light Detection and Ranging (LiDAR), and inertial measurement units (IMU's) systems are frequently utilized in robotics for motion tracking and spatial analysis, their application to high-accuracy tool trajectory evaluation in industrial machining remains constrained [14,15]. Laser trackers and LiDAR systems, which generate dense 3D point clouds by measuring laser reflections, can offer accurate spatial data under controlled conditions [16]. Additionally, the scanning frequency of most LiDAR units is insufficient to capture millisecond-scale deviations that occur during high-speed robotic operations, particularly along curved or segmented trajectories. Alternatively, IMU's provide motion estimates based on internal accelerometers and gyroscopes. Despite their lightweight nature, cost-effectiveness, and immunity to visual occlusion, IMU's are susceptible to accumulated drift, low spatial resolution, and an inability to directly measure absolute position in 3D space without sensor fusion [13,17]. The inherent limitations of the system under consideration were shown to impede the capacity to detect localized trajectory deviations with the requisite accuracy for contemporary robotic machining applications and calibration [18]. In such applications, errors of less than 1 mm can have substantial consequences for part quality. A more detailed comparison of high-speed camera, LiDAR, and IMU-based measurements is provided in Table 1 below.

Table 1. A comparison of sensor technologies for the use of robotic TCP tracking [14,15,17].

Feature	High-Speed Camera	Laser Tracking, LiDAR	IMU
Measurement Type	Visual position tracking	3D point cloud (distance)	Acceleration & orientation
Temporal Resolution	From medium (60–240 fps) to very high (1000+ fps)	Moderate (kHz range)	High, but noisy at high speeds
Spatial Accuracy	Sub-mm (with calibration)	mm-level, surface-dependent	Low without sensor fusion
Visual Feedback	Yes	Limited (requires processing)	No
Setup Complexity	Moderate	High	Low
Suitable for High-Speed Deviations	Yes	Limited	No

A High-speed motion camera system offers distinct advantages for real-time industrial robot TCP trajectory evaluation [19,20]. These systems can capture thousands of frames per second with sub-millimetre spatial resolution, enabling direct visual tracking of the robot's TCP without physical interference or reliance on internal estimations. In contrast to LiDAR or IMU systems, high-speed visual tracking facilitates detailed observation of dynamic deviations, such as overshoots, lags, or oscillations, especially during transitions or curved motion segments. The visual nature of the data renders it highly interpretable, facilitating rapid diagnosis of problems [21,22].

In ref. [23], a systematic approach for high-speed, high-accuracy spatial curve tracking in industrial robots was proposed by optimizing motion primitives in a simulation environ-

ment. It is acknowledged that performance objectives such as tracking accuracy, trajectory velocity, and motion uniformity frequently conflict [24]. The proposed method addresses the challenge of tuning coupled motion parameters, a process that is traditionally manual and time-consuming. The proposed optimization process involves selecting static parameters, choosing suitable motion primitives, and iteratively refining waypoints to minimize tracking errors. The objective of the research in ref. [23] is to maximize average trajectory velocity while adhering to specified tracking and speed uniformity constraints. The mentioned research and research provided in ref. [2] were focused on one of the motivations to evaluate a real robotic setup, since there are existing knowledge gaps in trajectory velocity and interpolation relations for different industrial robotics. It was acknowledged that the operational characteristics of industrial robots can vary depending on their respective kinematic and dynamic constraints, as well as the employed control strategies [25].

To make visual detection more convenient, some literature suggests using the Canny edge detection algorithm. This algorithm was implemented in industrial robot systems for tasks such as visual servo control, weld seam tracking, defect detection, and path planning, often using OpenCV or GPU-accelerated variants to ensure real-time performance [26].

To address the aforementioned limitations and research gaps, this paper proposes the use of a high-speed motion camera system with the Canny detection algorithm for visual, non-contact tracking of TCP trajectories. This method facilitates detailed observation of the end-effector's motion with temporal and spatial accuracy. By capturing the robot's TCP in real time and under actual trajectory tracking conditions, this technique enables a clearer understanding of how trajectory execution accuracy varies with velocity, trajectory geometry, trajectory interpolation, and robot kinematics. A notable advantage of this approach is its capacity to facilitate visual error monitoring. This capability enables researchers and engineers to not only quantify deviation magnitudes, but also visually interpret their distribution and cause. Conventional sensor-based techniques generally encounter difficulties when attempting this level of analysis.

The primary objective of this study is to establish a camera-based methodology for evaluating trajectory-following accuracy in industrial robots. The study suggests a methodology for the quantitative evaluation of industrial robot tool trajectories, with a particular focus on detecting deviations. This methodology utilizes imaging technology, such as a fast camera, to facilitate the analysis of tool trajectory deviations. The proposed framework is designed to be adaptable and suitable for a wide range of robotic arm models and motion primitives, including both linear and circular interpolation commands. The proposed approach is expected to foster a robust, data-driven understanding of robot motion behaviour, thereby enabling both academic investigation and industrial quality control in high-speed robotic additive manufacturing applications, like uniform material deposition or distribution.

2. Materials and Methods

This suggested approach is designed to be adaptable across all types of industrial articulated robots performing tasks that utilize trajectory motion planning. Additionally, it takes into account motion types, including both linear and nonlinear types, referring to first-degree and second- or higher-degree polynomials under varying speed conditions. The proposed methodology is designed to facilitate precise and repeatable performance evaluation, enabling comparative benchmarking, error analysis, and the development of compensation strategies in various industrial and research applications.

2.1. Rationale of the Research

Trajectory interpolation is defined as the process by which a robot controller determines the coordinates of intermediate set points for the robot's TCP to follow, executing motion commands that define only a few waypoints. The quality of interpolation is a quantitative metric that quantifies the extent to which the robot's trajectory closely resembles the intended trajectory. It is calculated by measuring the smoothness of the motion, which refers to how continuous the robot's tool movement is over time, without any sudden jerks or discontinuities in velocity or acceleration, over the trajectory [27,28]. Low-quality interpolation, defined as coarse or simplistic interpolation, can result in trajectory deviations and positioning errors [2].

The evaluation of the influence of linear interpolation strategies on the accuracy and continuity of tool trajectory generation in industrial robotics is a primary interest of this research. In numerous robotic machining and additive manufacturing applications, discretizing trajectories into linear segments is a conventional practice; however, it introduces approximation errors, especially in curved or complex trajectories. As illustrated in Figure 1A–D, varying interpolation resolutions (or tolerances) introduce geometric deviations. As demonstrated, increasing the segment length from 4.624 mm (Figure 1B) to 9.593 mm (Figure 1D) results in visibly larger angular transitions and reduced surface smoothness, highlighting the trade-off between computational efficiency and geometric fidelity. It is essential to quantitatively assess these interpolation-induced errors to optimize robot motion planning, minimize edge or surface imperfections, and ensure consistent process quality, particularly in applications where high-speed and high-accuracy trajectory following are required.

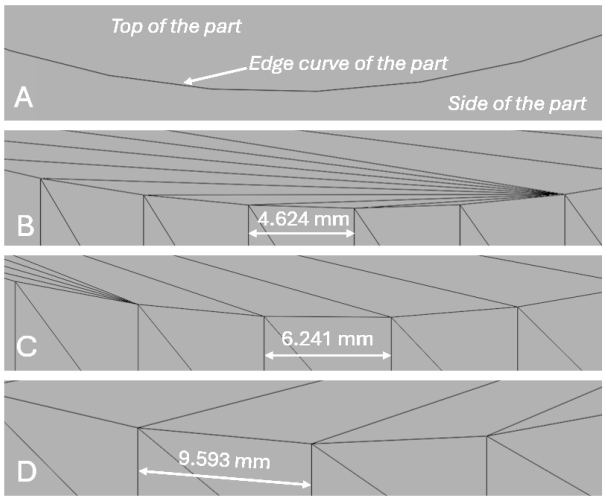


Figure 1. Visual representation of the tolerance (T) influence: (**A**) original part; (**B**) T = 0.05 mm; (**C**) T = 0.1 mm; (**D**) T = 0.25 mm.

The research methodology was designed to systematically evaluate and compare the trajectory-following accuracy of industrial robots under two different trajectory interpolation strategies. The schematic representation of the pipeline is given in Figure 2. This pipeline represents two different interpolation strategies for different trajectories, and both are implemented in parallel on both industrial robots. The first strategy is first-degree (linear) interpolation with adjustable tolerances, and the other strategy is second-degree (circular) interpolation using special motion commands. The general objective was to determine how these interpolation types, in combination with different velocities, influence the robot's ability to follow predefined trajectories accurately.

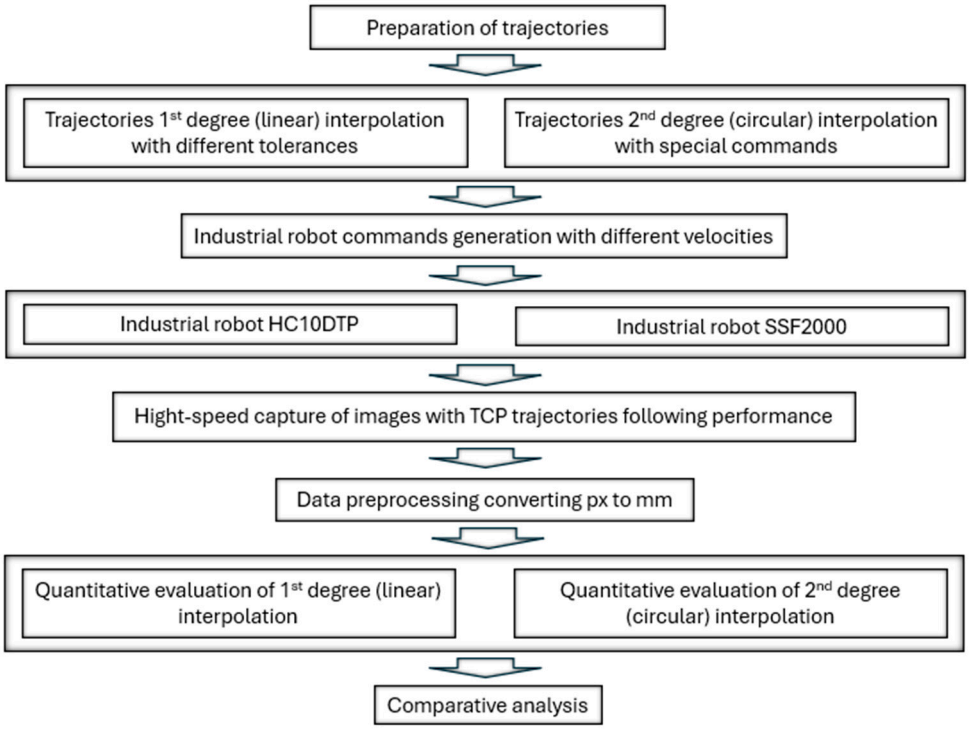
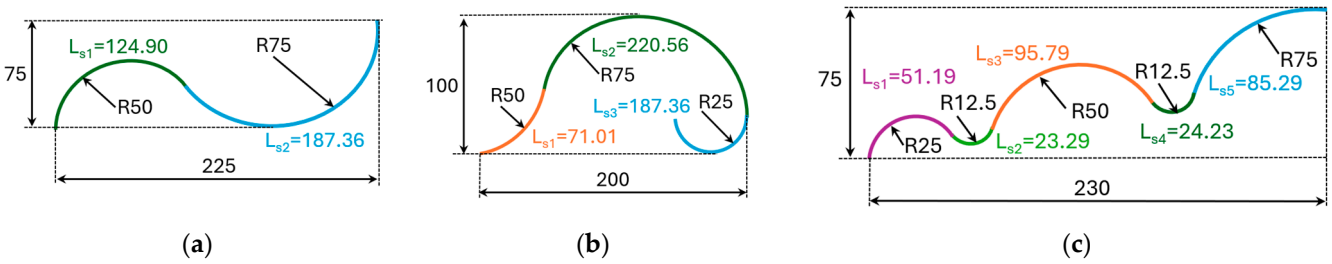


Figure 2. The research methodology pipeline.

Two types of trajectory data were generated using linear interpolation with varying tolerance values to test segment resolution, and second-degree circular interpolation using robot circular motion commands to allow the robot controller to approximate arcs and curves without straight-line segmentation. Following the definition of the trajectory data, it was imported into the "RoboDK" (v5.8.0, RoboDK, Montreal, QC, Canada) simulation environment to generate motion commands specific to the robots. These commands were configured with various velocity profiles to enable the assessment of interpolation performance across different motion speeds. The programs thus prepared were subsequently deployed to two distinct industrial robots: the "Motoman HC10DTP" (Yaskawa, Kitakyushu, Japan) and the "Motoman SSF2000" (Yaskawa, Kitakyushu, Japan), which were selected based on their differing age, usage levels, and controller architectures. It is known that the performance of industrial robots depends on their manufacturer and controller software platforms [25].

Three distinct trajectories (Figure 3) were defined, each characterized by a unique curvature radius of the arcs that comprise said trajectory. In this research, trajectories based on the radius of curvature are used to create a unified and quantitative model for evaluating the quality of robot motion. Representing trajectories according to their radius of curvature allows for the consistent evaluation of both linear and curvilinear motions within the same model. This method simplifies motion analysis, helps evaluate motion smoothness, and ensures comparable results across different trajectory geometries. Figure 3 depicts the aforementioned trajectories, which were randomly defined in terms of their dimensions and the points of intersection between the arcs. The concept of trajectory shapes is based on the research article under ref. [29]. Each arc is characterized by separate sections with specific radius and length within different trajectories. The first trajectory (Figure 3a) represents a simple transition from one arc shape to another. The second trajectory (Figure 3b) is intended to represent multiple radius segments with more dynamic transitions between segments, and the third trajectory (Figure 3c) is intended to represent a composite trajectory

featuring tighter radius and multiple curvature transitions, representing a more complex motion profile.



(**a**) (**b**) (**c**)

Figure 3. Sketches of different trajectories for the experiments: (**a**) first trajectory; (**b**) second trajectory; (**c**) third trajectory.

Subsequently, the trajectories were designed using "SolidWorks" (2024, Dassault Systèmes, USA, Concord, MA, USA) software, which permitted the control of tolerances (T) for the generation of STL model files and the implementation of linear interpolation between the points of the aforementioned trajectories. Table 2 illustrates the influence of tolerances and their impact on the number of data points and the distance between those points across different trajectories. Trajectories with varying tolerances, ranging from 0.05 to 0.5 mm, were uploaded into the "RoboDK" (v5.8.0, RoboDK, Montreal, QC, Canada) software environment for further preparation of the robot code generation process.

Table 2. The impact on the number of data points within certain tolerances.

Trajectories	**1st**		**2nd**			**3rd**				
Length of trajectories (mm)	312.26		370.11			279.79				
Trajectory segment number i	1st	2nd	1st	2nd	3rd	1st	2nd	3rd	4th	5th
Length segments L_{si} (mm)	124.90	187.36	71.01	220.56	78.51	51.19	23.29	95.79	24.23	85.29
Number of points (pieces) by different tolerances T (mm)										
T = 0.05	27	33	16	41	25	17	11	22	11	16
T = 0.10	20	25	12	30	18	12	8	16	8	11
T = 0.25	13	16	8	18	11	8	5	10	5	7
T = 0.50	7	8	4	9	6	4	3	5	3	4
Distance between points (mm) by different tolerances T (mm)										
T = 0.05	4.6240	5.6760	4.4370	5.3780	3.1395	3.0090	2.1150	4.3530	2.1995	5.3300
T = 0.10	6.2410	7.4910	5.9140	7.6020	4.3580	4.2605	2.9050	5.9830	3.0210	7.7510
T = 0.25	9.5930	11.6980	8.8650	12.2395	7.1160	6.3810	4.6320	9.5650	4.8150	12.1720
T = 0.50	19.0984	23.3242	17.6597	24.3974	14.0870	12.6579	9.1030	19.0415	9.4500	24.2630

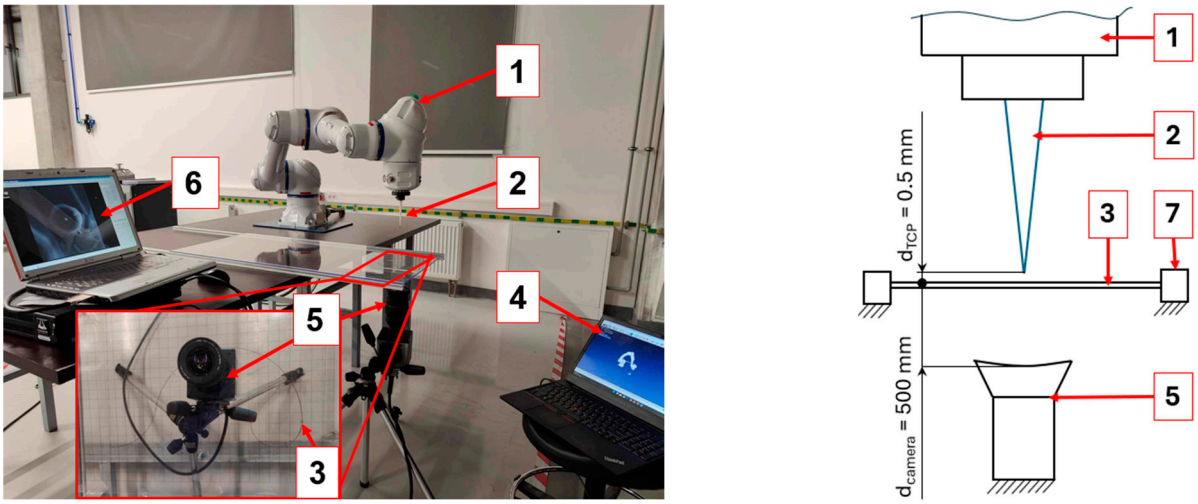
Each robot executed the programmed trajectories, during which a high-speed camera system recorded the TCP motion. This setup enabled non-contact, high-resolution, and high-speed tracking of the robot's motion trajectories, allowing for visualization and quantitative analysis of deviations from the reference trajectory. The captured video frame data were pre-

processed to convert image coordinates (pixels) to real-world units (mm) using a calibration procedure. This facilitated an accurate spatial comparison between the commanded and attained trajectories. Subsequently, the recorded motion data were analyzed separately for each type of interpolation. For the evaluation of the linear interpolation part, trajectory-following errors were evaluated across various tolerances and velocities. For the evaluation of the second degree (circular) interpolation part, a similar evaluation was conducted to assess how well the robot controller handled curvature and transitions without explicit linear segmentation.

The final step involved a comparative analysis to interpret the differences in performance between the two interpolation strategies. Key metrics included average velocity deviation, positional deviation, and trajectory accuracy under varying conditions. This comprehensive approach is yielding valuable insights into the behaviour of industrial robots under different trajectory-generation schemes and providing insight into strategies for enhancing accuracy in high-speed robotic applications.

2.2. Experimental Setup

The experiments were conducted using two industrial robots from the "Yaskawa" manufacturer as a test bench (Figure 4) for industrial robotics. The first robot was "HC10DTP" (Yaskawa, Kitakyushu, Japan) (Figure 4, position 1), with "YRC1000" (Yaskawa, Kitakyushu, Japan) controller, and operated totally up to 100 h; the second robot was "SSF2000" (Yaskawa, Kitakyushu, Japan), with "NX100" (Yaskawa, Kitakyushu, Japan) controller, and operated totally up to 1600 h.



(**a**) (**b**)

Figure 4. Experimental setup: (**a**)—picture of setup; (**b**)—scheme of imaging; 1—industrial robot "HC10DTP" (Yaskawa, Kitakyushu, Japan), 2—industrial robot tool, 3—transparent foil sheet with test trajectory, 4—PC with running "RoboDK" (v5.8.0, RoboDK, Montreal, QC, Canada) software, 5—"FastCam 1024 PCI" (Photron, Yonezawa, Japan) camera, 6—PC with camera software; 7—frame for foil sheet.

In both research cases, robots and reference trajectories were fixed to a custom-designed, stable base, ensuring sufficient mounting stiffness and a reliable relation between the robot base and the reference trajectory. The high-speed camera was mounted separately on the ground. Such a setup ensures the reliable measurement of robot TCP deviation from the reference trajectory and partially eliminates the impact of vibrations caused by robot motion. Only the relative position between the corresponding pixels in the image is taken into account.

Table 3 presents the geometric parameters and main characteristics of these robots, and additionally, Tables 4 and 5 show Denavit–Hartenberg (DH) parameters of two industrial robots. The needle-shaped tool with a marked tip (Figure 4, position 2) is attached to the robot arm, which is calibrated to follow a trajectory. The needle tool is designed for the accurate calibration of a tool and for enabling easy distinguishable tracking of the tip point. The weight of the toll is approximately 0.1 kg. The reference trajectory patterns (Figure 4, position 3) were printed on transparent foil sheets and placed between the robot's working plane and the camera.

Table 3. Main characteristics of industrial robots used in the research [30].

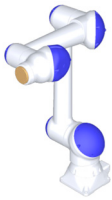
Figure of Robot	Model	Number of Axes	Payload	Reach	Repeatability	Weight
	"HC10DTP"	6	10.0 kg	1200 mm	0.1 mm	47 kg
	"SSF2000"		6.0 kg	1378 mm	0.08 mm	130 kg

Table 4. DH parameters of industrial robot "HC10DTP" [30].

Joints	α (deg)	a (mm)	θ (deg)	d (mm)
J1	0.0	0.0	0.0	275.0
J2	−90.0	0.0	−90.0	−3.0
J3	0.0	700.0	0.0	0.0
J4	−90.0	0.0	0.0	500.0
J5	90.0	0.0	0.0	162.0
J6	−90.0	0.0	180.0	170.0

Table 5. DH parameters of industrial robot "SSF2000" [30].

Joints	α (deg)	a (mm)	θ (deg)	d (mm)
J1	0.0	0.0	0.0	450.0
J2	−90.0	150.0	−90.0	0.0
J3	0.0	570.0	0.0	0.0
J4	−90.0	155.0	0.0	640.0
J5	90.0	0.0	90.0	0.0
J6	−90.0	0.0	0.0	95.0

The DH parameters offer a standardized approach to the description of the geometry of a serial robot manipulator [31]. The configuration of each joint-to-joint link is defined by a set of four parameters. The parameter θ_i denotes the joint angle, i.e., the rotation

around the *z*-axis; d_i represents the link offset, i.e., the translation along the *z*-axis; θ_i stands for the length of the link (translation along the *x*-axis); and α_i is the link twist (rotation around the *x*-axis). Utilizing these, a sequence of homogeneous transformation matrices was constructed, extending from the robot base to the TCP.

In a nutshell, according to the tables above and the explanation of DH parameters relationships in [31], both systems share the same twist topology, but "SSF2000" spreads length across multiple links and sits higher, thus favouring broader, better-conditioned mid-workspace motion. In contrast, "HC10DTP" concentrates reach in one long forearm with larger wrist offsets, which can heighten sensitivity to compliance/backlash at extension and affect high-speed TCP tracking.

The development of all robot programs was conducted using "RoboDK" (v5.8.0, RoboDK, Montreal, QC, Canada), a widely adopted offline programming and simulation platform for industrial robots (Figure 4, position 4). Its integration of real-world robot controllers facilitates the consistent use of virtual trajectories on physical robots, reducing development time, providing versatility, and increasing programming accuracy in terms of how closely the real robot follows the task. This is defined in the simulation setup. The experimental trajectories were generated with precise spatial control and exported directly to the robot controllers, ensuring consistency across all test runs.

In order to visually observe and detect any deviations in the performed trajectory of the robot tool that follow the real trajectory, the "FastCam 1024 PCI" (Photron, Yonezawa, Japan) high-speed imaging camera Figure 4, position 5) was employed. This camera is capable of operating from 60 to 1000 frames per second (fps) at a full resolution of 1024 × 1024 pixels or up to 3000 frames per second at a full resolution of 512 × 512 pixels, for reference. Additionally, this camera has 10-bit CMOS with a large, light-sensitive pixel sensor and a 1.5-microsecond electronic global shutter. The "Photron FASTCAM Viewer 3" (Photron, Yonezawa, Japan) software (Figure 4, position 6) was used for the preparation of the camera setup and the initial pre-processing of the visual data, adjusting the contrast and the brightness. The calibration checkerboard was implemented to properly align the camera view with the observable reference trajectories on the transparent foil using similar methodologies as provided in ref. [32].

2.3. Experimental Procedure

The initial preparation stage involved the automatic generation of robot code for executing the trajectories. This was achieved by generating linear-type movements, which resulted in a rapid programming process. Another type of interpolation, specifically for circular shape movements, was implemented in parallel with linear interpolation. All the configurations with three different linear velocities of a tool were programmed at 50 mm/s, 250 mm/s, and 500 mm/s. This was performed to observe the influence of each trajectory segment with its corresponding tolerance on achieving the true value. The selection of these velocity sizes was made in accordance with the standards outlined in ISO 9283:1998, a document that provides guidelines for the manipulation and performance evaluation of industrial robots [33]. This standard proposes the use of relative velocities corresponding to 10%, 50%, and 100% of the maximum velocity used in this research.

To ensure the accuracy of the robot's end effector or TCP positioning and movement, it was essential to perform a calibration process of the tool position. The calibration process typically involves determining the precise location of the TCP with respect to the centre of the robot arm end flange. This necessitates the movement of the tool through a series of predefined points in space, employing a technique such as the "five-point" method to ascertain the precise TCP for "Motoman" (Yaskawa, Kitakyushu, Japan) robots. Additionally, other manufacturers may utilize the "four-point" or "six-point" methodologies to calculate

the TCP. In order to ensure precise alignment of the two-dimensional trajectory plane with the reference trajectory, it was necessary to define the user coordinate system (Figure 5).

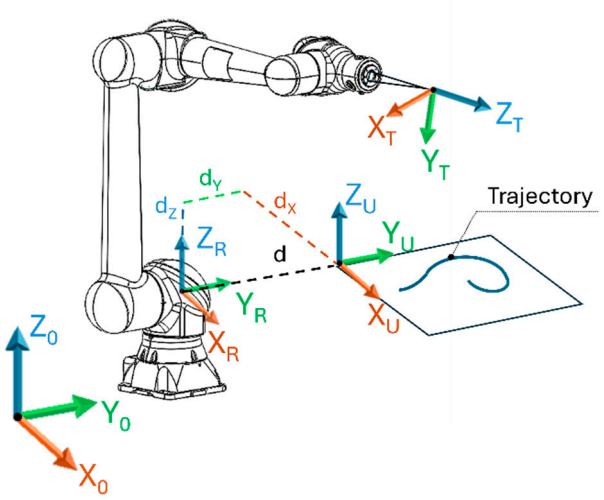


Figure 5. Different coordinate systems in industrial robot setup: X_U, Y_U, and Z_U—user coordinate system; X_0, Y_0, and Z_0—global coordinate system; X_R, Y_R, and Z_R—robot coordinate system; X_T, Y_T, and Z_T—TCP coordinate system; d—distance between robot coordinate system and user coordinate system.

The user coordinate system (Figure 5, X_U, Y_U, Z_U) is typically established in relation to reference points that represent critical locations or features of the workspace or the task environment. This allows the robot's movements to be described in relation to the specific geometry of the task at hand, rather than relying solely on a global (Figure 5, X_0, Y_0, Z_0) or robot coordinate system (Figure 5, X_R, Y_R, Z_R). The user coordinate system thus corresponds directly to the physical layout of the work environment (in our case, a transparent foil sheet with a test trajectory).

Furthermore, the user coordinate system can be modified in real-time as the task evolves. To illustrate, if the robot is engaged in different phases of complex tasks, the coordinate system can be redefined based on new reference points at each stage. This ensures the robot's ability to adapt to the varying regions of the workspace. In our experiment, the technique enabled us to align the industrial robot's TCP (Figure 5, X_T, Y_T, Z_T) movement with the printed trajectory patterns with exact accuracy and easily define several relative position changing coordinates (Figure 5, d_x, d_y, d_z) with respect to the robot coordinate system.

To monitor the trajectory performance accuracy, the "FastCam 1024 PCI" (Photron, Yonezawa, Japan) was configured to record data at a resolution of 512 × 1024 pixels and 500 fps. This configuration was chosen to ensure a resolution of 2 milliseconds, thus facilitating the observation of TCP displacement of 0.1 mm at 50 mm/s velocity. For an accurate evaluation of the distance, it was ascertained that a distance of 10 pixels in a given frame was equivalent to 1 mm. The vertical distance from the TCP to the transparent foil sheet was approximately d_{TCP} = 0.5 mm, and this distance was not considered in the accuracy calculation due to its negligible influence on errors. The distance from the focal length of the camera to the transparent foil sheet was d_{camera} = 500 mm (Figure 4b). In order to avoid potential vibrations of the tool tip, the measurements of motion performance along the trajectory were taken after the settlement of the motion.

In addition, the Canny edge detection algorithm was used to improve the analysis of trajectory accuracy and TCP deviation. By applying Canny-based edge detection to visually collected data, it became possible to identify contour deviations automatically and local irregularities along the robot's trajectory. This method allowed for a reliable

comparison of the specified and actual trajectories with a recognition accuracy of 96.3%, facilitating a quantitative assessment of the accuracy of interpolation and the efficiency of trajectory tracking. The detection procedure was, in parallel, similar to the recent research in [26]. In short, firstly, the acquired image data were pre-processed by converting them to grayscale and applying Gaussian smoothing to reduce noise and ensure stable edge detection. Then, the Canny method utilizing the OpenCV library was used to calculate image gradients and identify high-contrast transitions that correspond to the edges of the trajectory. Additionally, separate frames were created to detect the robot TCP real path. Finally, two separate frames containing information about the detected trajectories and TCP numerical information were merged into one, and then the deviations were calculated.

It is important to note that this research does not involve the evaluation of detailed velocity profiles; therefore, only the average velocity for each trajectory segment is considered. The average velocities and deviations were determined, which are provided in the Section 3, by performing and monitoring five independent measurements of the robot's movement between pairs of trajectory points using both linear and circular interpolation modes. More detailed information about the measurement values can be found in the Supplementary Materials. This methodological approach ensured statistical consistency and minimized random deviations caused by controller dynamics or measurement noise.

2.4. Statistical Evaluation

The overall positioning accuracy of the robot end effector was evaluated by performing axis-specific measurements, according to the standardized methodology defined in ISO 9283:1998 [33], which describes the performance criteria and test procedures for industrial robots. According to this standard, positioning accuracy AP_p is quantified as the deviation between a reference point in space and the average of several achieved positions, each achieved by approaching the target in the same direction. This directional consistency helps to isolate positioning inaccuracies related to mechanical repulsion, control resolution, and interpolation errors. Mathematically, the ISO 9283 standard defines positioning accuracy AP_p as

$$AP_p = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2}, \quad (1)$$

where x_c, y_c, z_c are the coordinates of the commanded prescribed end-of-arm TCP reference position in the robot's coordinate system, and $\bar{x}$, $\bar{y}$, $\bar{z}$ are the mean values of the actual attained positions along each respective axis, calculated over multiple repetitions.

The definition of trajectory accuracy is independent of the shape of the command trajectory. The trajectory characteristics that are described in the following sections are generally applicable to all programming methods. As illustrated in Figure 6, this is a general representation of trajectory accuracy.

According to ISO 9283:1998 [33] and some experimental measurements of a robot's path accuracy in ref. [34], trajectory accuracy is defined as the degree to which a robot's TCP aligns with a designated trajectory during uninterrupted movement. The figure illustrates this by comparing the attained path to the command path at each reference point P = (X_c, Y_c, Z_c). A normal plane is constructed perpendicular to the command (or target) path at each segment, and the shortest distance from the attained trajectory to this plane is recorded as the path deviation AP_p. This approach captures the robot's dynamic tracking accuracy, not just point-to-point accuracy, making it particularly relevant for processes requiring smooth, uninterrupted motion, such as machining or welding. The method enables a standardized evaluation of trajectory-following performance, highlighting deviations caused by system dynamics or control limitations.

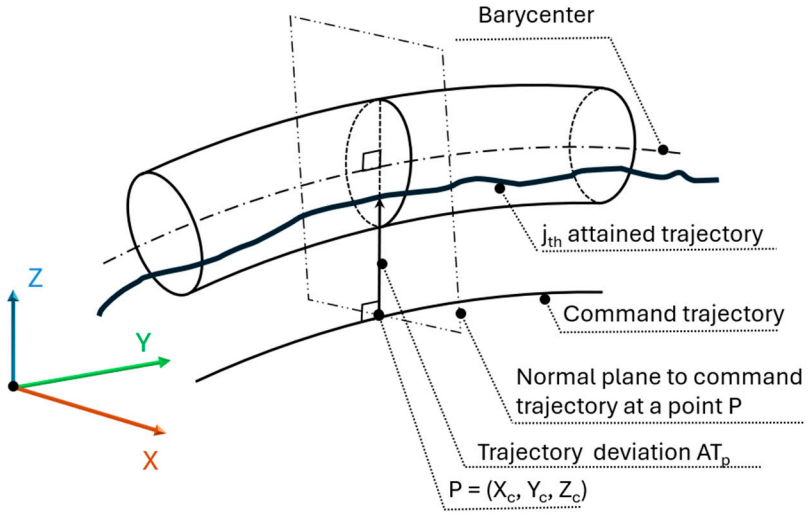


Figure 6. Trajectory accuracy for a command trajectory representation based on ref. [30].

For the experimental evaluation of general trajectory positioning accuracy error over several operation times, measurements can be performed along all three Cartesian axes (X, Y, Z). However, in this particular scenario, only two axes of measurements (2D) are performed as information is acquired with the high-speed camera. Since the target position is defined as $x_c = 0$, $y_c = 0$, and $z_c = 0$, the deviation is calculated as follows:

$$AP_p = \sqrt{\left(\overline{x}\right)^2 + \left(\overline{y}\right)^2 + \left(\overline{z}\right)^2}\,. \tag{2}$$

But eliminating the $\overline{z}$ variable, a simplified form of the ISO 9283 positioning accuracy equation becomes as follows:

$$AP_p = \sqrt{\left(\overline{x}\right)^2 + \left(\overline{y}\right)^2}\,, \tag{3}$$

where $\overline{x}$, $\overline{y}$ are calculated from experimental data as follows:

$$\overline{x} = \frac{1}{n}\sum_{j=1}^{n} x_j,\ \overline{y} = \frac{1}{n}\sum_{j=1}^{n} y_j, \tag{4}$$

where x_j, y_j are the coordinates of the resulting position, and n is the measurement cycle number.

This simplification reflects the assumption that all commanded target positions are at the origin, allowing for a direct calculation of the magnitude of the displacement vector from the ideal point. This metric offers a unique scalar value that represents the overall spatial deviation, which is particularly useful for quantifying degradation or drift over time, as well as for evaluating the effectiveness of proposed compensation strategies.

3. Results

3.1. Experimental Evaluation of "HC10DTP" Robot TCP Accuracies

The experiment initially focused on assessing the industrial robot's motion capabilities, with a particular emphasis on the accuracy and reliability of its TCP movements along different trajectories. To ensure a controlled and quantifiable evaluation, the TCP movement was initiated using the least sophisticated form of linear interpolation. This approach was selected to establish a baseline for system performance. The objective was to capture the maximum possible deviations from the reference trajectory and the influence on the robot's TCP velocity based on the distance between sequential points. The initial experiment's

results are presented in Figure 7, which utilizes different radius values to represent each trajectory segment, corresponding to the programmatically set velocities and the evaluated actual velocities. Moreover, the average velocities throughout the whole trajectories are documented in the legends.

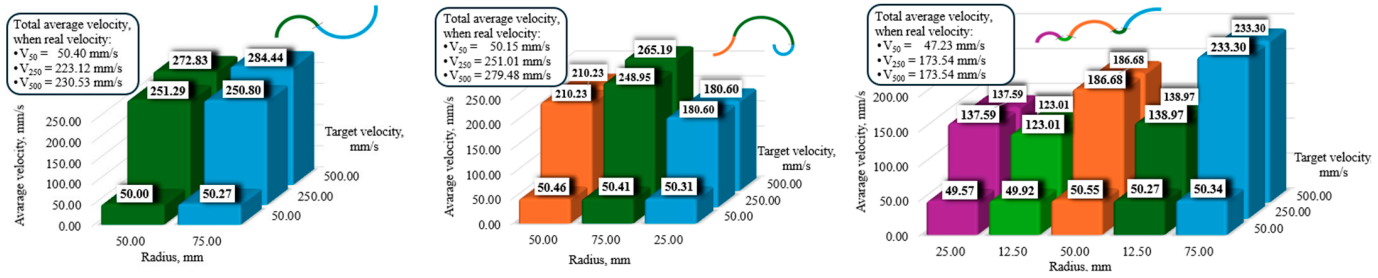


Figure 7. Average velocity between points with "HC10DTP robot, when T = 0.5, here (**a**) of the first trajectory; (**b**) of the second trajectory; (**c**) of the third trajectory.

As illustrated in Figure 7, the trajectory of TCP performance, when the target velocity is fixed at 50 mm/s, closely follows the expected magnitude in most scenarios. However, as the target velocity increases, only trajectory segments with a greater curvature radius can reach the predefined velocity. This effect is primarily due to the motion controller's built-in velocity constraints, which prevent excessive deviations from the programmed trajectory and ensure smooth motion transitions. Additionally, the ability to achieve the target velocity is strongly influenced by the length of the arc segment. Shorter arc segments, especially those with tight curvature, limit the time available for acceleration, resulting in the actual velocity remaining lower than the commanded velocity. As a result, the overall influence on performance time is decreased.

Given these observations, in order to evaluate how interpolation accuracy behaves under demanding motion conditions, a target velocity of 250 mm/s was chosen. For reference purposes, as illustrated in Figure 8, the visual monitoring of the TCP deviation under investigation was conducted using a high-speed camera. This monitoring was performed in the third segment of the third trajectory, under various conditions.

This choice enables a more detailed examination of how interpolation accuracy impacts motion dynamics at higher velocities, which play a crucial role in trajectory execution. Further, Figure 9 presents the average velocity between points and the deviation from different trajectories with the corresponding tolerance sizes. Figures present a comparative analysis of three different trajectory geometries, each divided into segments of different radius, and evaluate the influence of interpolation tolerance on both the average velocity between points and the resulting deviation from the intended trajectory. This analysis is relevant in contexts that require precise high-speed motion, such as robot tool trajectory planning.

The upper row of graphs in Figure 9 displays the average speed achieved between interpolation points at four different tolerance values: 0.05 mm, 0.1 mm, 0.25 mm, and 0.5 mm. A consistent trend is noticeable across all trajectory types in the data. It is evident that better tolerances (ranging from 0.05 mm to 0.1 mm) yield the lowest average velocities, which frequently fall within the range of 40–100 mm/s in most scenarios. This can be attributed to the increased density of interpolation points. This increased number of points requires the system to slow down frequently to maintain accurate tracking, especially in segments with a smaller radius. As the tolerance number is increased (ranging from 0.25 mm to 0.5 mm), the average velocities tend to approach the target value of 250 mm/s. At a tolerance of 0.5 mm, velocities are found to be close to or at the target, particularly on trajectories with larger radius or simpler curvature profiles. Additionally, the segment arc

length also influences the magnitude of velocities. For instance, in Figure 9b, the distance between points in the segment radius of 75 mm is 1.05 times longer than the same radius segment in Figure 9a, and 1.005 times longer than the same radius segment in Figure 9c. A comparable tendency ratio is observed between Figure 9b,c geometries, exhibiting velocity ratios ranging from approximately 1 to 1.1 times. A similar trend is noted in the ratio between Figure 9b,c segments, which approaches 1.1 times when the tolerance is set at 0.25 and 0.50, and approaches 0.85 when the tolerance is set at 0.05 and 0.1.

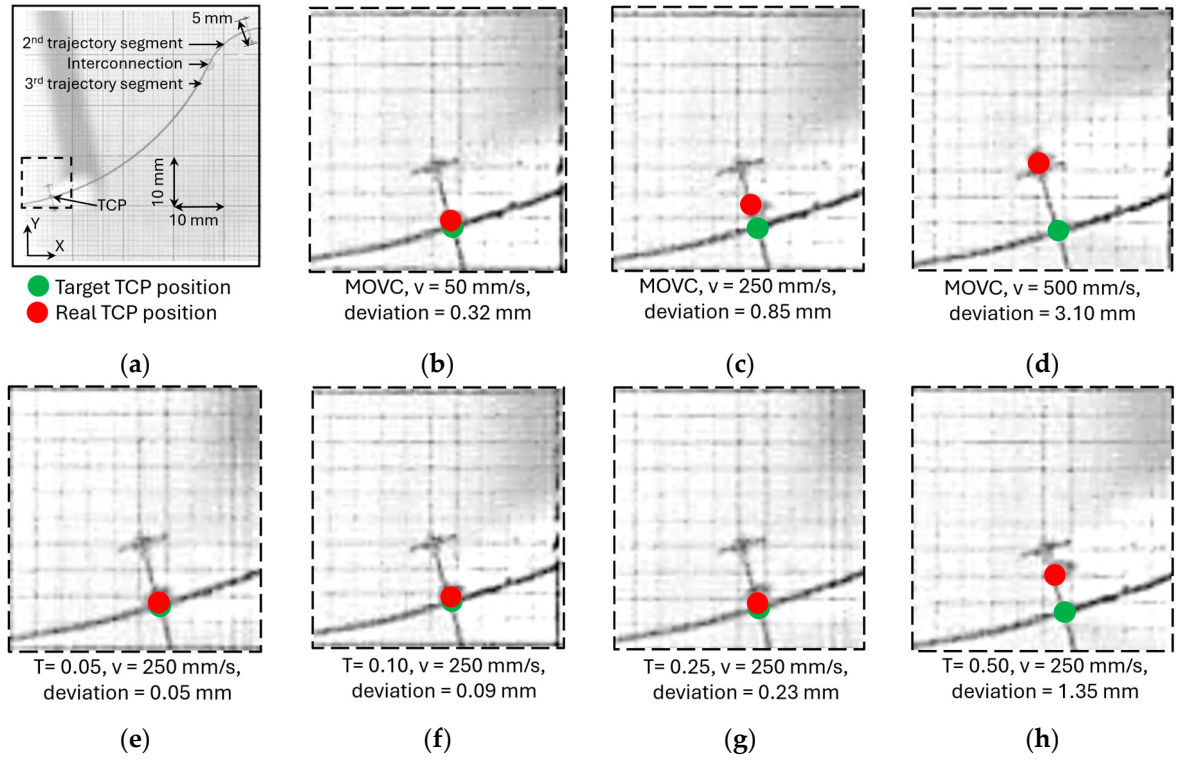


Figure 8. TCP following monitoring under the high-speed camera in the third segment of the third trajectory with various conditions, here (**a**) zoomed image from the high-speed camera; (**b**–**h**) observations of TCP deviations in the presence of different conditions.

The overall average velocity, shown in the label boxes of each graph, also confirms this trend. For instance, in the first trajectory, the average velocity increased from 76.85 mm/s with a tolerance of 0.05 mm to 251.09 mm/s with a tolerance of 0.5 mm in Figure 9a geometry. The rate of increase in average velocity displays a nonlinear pattern. This observation indicates that the algorithm prioritizes the smoothness of motion over strict geometric trajectory tracking at higher tolerances. This prioritization was demonstrated to result in a substantial reduction in computational overhead and motion delay.

The graphs in the bottom row are a quantification of the deviation from the ideal trajectory over the same tolerance levels and radius. This particular aspect of the analysis is crucial in applications where trajectory accuracy is most important. Minimal deviations, irrespective of the radius ($\leq$0.1 mm), are observed for tolerances of 0.05 mm and 0.1 mm. These tolerances guarantee that the interpolated trajectory maintains close alignment with the original trajectory, even in sections characterized by high curvature. However, at 0.25 mm and especially at 0.5 mm, the deviation increases significantly, reaching 0.99 mm or more. This increase in deviation is most noticeable for trajectory segments where the distance between consecutive points is longer, where the simplification of the interpolation leads to a significant deviation from the reference trajectory. The increase in deviation is found to be relatively uniform across different radius values, indicating that the interpo-

lation method introduces a global error proportional to the allowed tolerance and is not solely dependent on curve sharpness.

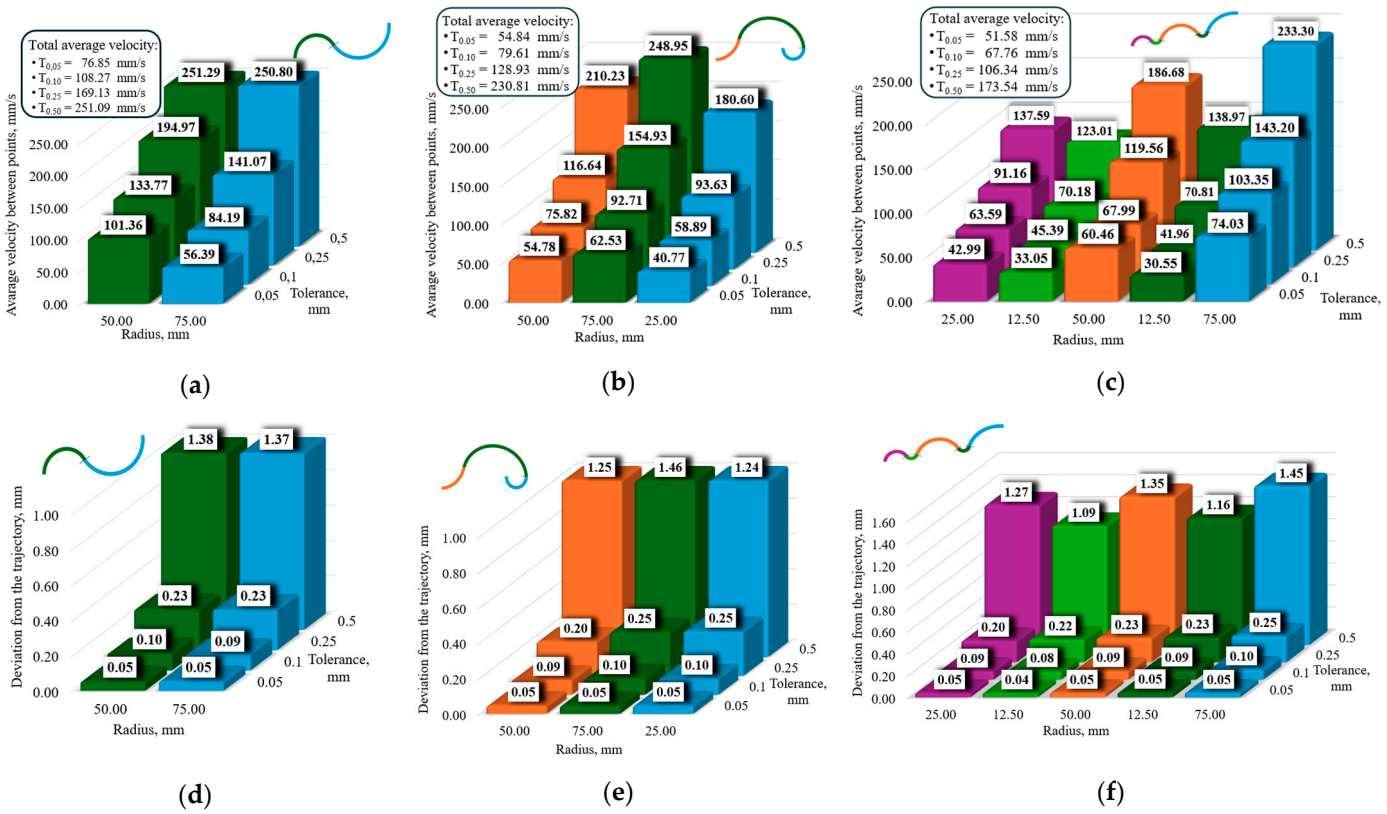


Figure 9. Average velocity between points and deviation from the trajectory implementing linear interpolation motion commands using "HC10DTP" robot, here (**a**) average velocity values for the first trajectory; (**b**) average velocity values for the second trajectory; (**c**) average velocity values for the third trajectory; (**d**) deviations in accordance with the first trajectory; (**e**) deviations in accordance with the second trajectory; (**f**) deviations in accordance with the third trajectory.

In another part of observations in Figure 10, to evaluate what is the influence of special industrial robotic motion commands, such as circular shape motion, on the trajectory following accuracy under several target velocities, these velocities are 50 mm/s, 250 m/s, and 250 mm/s. Further, Figure 10 presents average velocities and deviation from the trajectory using circular motion commands.

As illustrated in the upper row of Figure 10, the average velocity distributions across the trajectory segments executed with separate circular commands are shown. At the target velocity of 50 mm/s, the average velocity remains stable and close to the target across all trajectory configurations, with most values ranging from 44.99 mm/s to 49.97 mm/s. This outcome suggests that there are minimal interruptions in motion execution at low velocities. It was also observed that there is a short-time correction process at key points defining the circular/arc shape trajectory, which is about 0.1 s. At the nominal speed of 250 mm/s, performance varies by geometry. The average velocity ranges from 155 mm/s to 230 mm/s, with tighter and shorter curves reducing the velocity more significantly. For instance, in the second trajectory (with a radius of 25 mm), the velocity decreases to about 155 mm/s, while segments with a larger radius (e.g., 75 mm) reach more than 190 mm/s. At the maximum velocity of 500 mm/s, average velocities attain peaks exceeding 280 mm/s in specific segments. However, numerous segments fall within the range of 180–230 mm/s, which remains substantially below the targeted velocity. This observation suggests the presence of velocity limitations and restrictions resulting from the complexity of the dynamic trajectory.

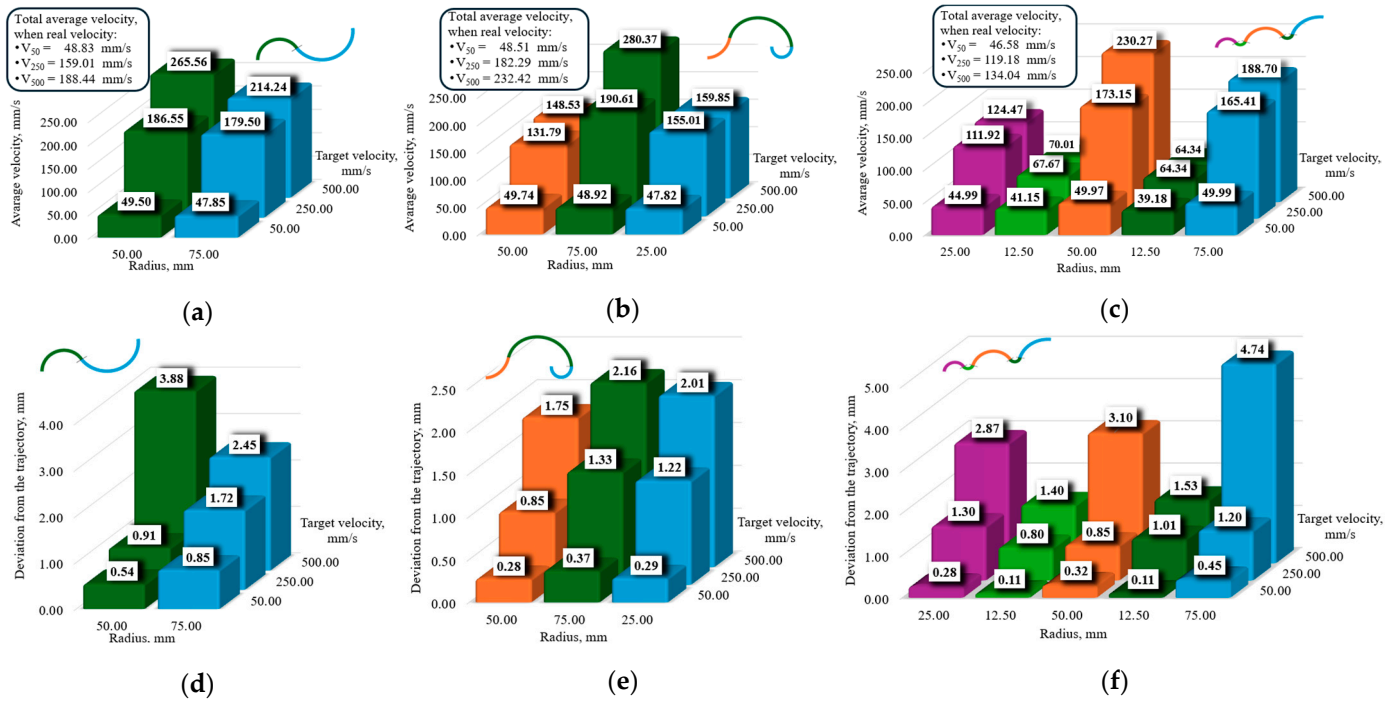
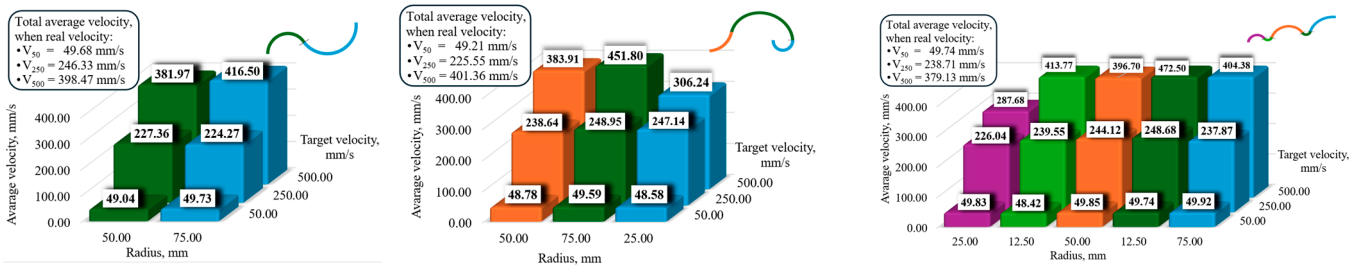


Figure 10. Average velocities and deviation from the trajectory implementing circular motion commands using "HC10DTP" robot, here (**a**) average velocity values for the first trajectory; (**b**) average velocity values for the second trajectory; (**c**) average velocity values for the third trajectory; (**d**) deviations in accordance with the first trajectory; (**e**) deviations in accordance with the second trajectory; (**f**) deviations in accordance with the third trajectory.

As illustrated in the bottom row of Figure 10, the robot's actual motion deviated from the ideal trajectory under different speeds. At a velocity of 50 mm/s, the deviation remains well controlled. The maximum deviations recorded range from 0.32 mm to 0.91 mm, often less than 1 mm, even for greater radius curve segments. This performance is comparable to or slightly decreases the trajectory accuracy observed with linear interpolation at 0.1 mm tolerance. However, as the velocity increases to 250 mm/s, a significant increase in trajectory deviation is observed. Some values exceed 2 mm, particularly in greater radius trajectories. For instance, in the second trajectory, deviation reaches 2.45 mm for a radius of 75 mm, and in the third trajectory, values such as 2.87 mm (R = 25 mm) and 3.10 mm (R = 50 mm) are observed. At an increased target velocity of 500 mm/s, the deviations become more prominent. The maximum recorded deviation was 4.74 mm in the 75 mm segment of the third trajectory. This observation signifies the heightened challenge in preserving accuracy at higher velocities, which may be due to inertia and the limitations of the controller.

3.2. Experimental Evaluation of "SSF2000" Robot TCP Accuracies

Another industrial robot demonstrated a strong alignment with the target velocities across larger radius segments, evaluating only with a lowest tolerance for linear interpolation of T = 0.5 mm, as well (Figure 11). At the maximum target velocity of 500 mm/s, the robot attained 381.97 mm/s and 416.50 mm/s for a radius of 50 mm and 75 mm, respectively. This suggests effective execution of motion. At a target velocity of 50 mm/s, the robot exhibited average velocities of 49.04 mm/s and 49.73 mm/s, which closely matched the target, thereby sustaining the system's reliability at low speeds. These findings underscore the controller's capacity to compensate for coarsely segmented trajectories, especially when motion complexity is moderate.



(**a**) (**b**) (**c**)

Figure 11. Average velocity between points with the "SSF2000" robot, when T = 0.5, here (**a**) of the first trajectory; (**b**) of the second trajectory; (**c**) of the third trajectory.

In other trajectories, the tendency is close to the first mentioned above; the only difference is visible in the radius of 25 mm, where there is a struggle to reach the target velocity better compared to other radii. These findings confirm that, even under coarse interpolation, the robot can preserve velocity integrity better compared with the robot before. However, potential positional inaccuracies or motion overshoots must be addressed through trajectory deviation analysis, as was previously observed in the evaluation of industrial robots beforehand. Therefore, further analysis is performed in the same manner as during the above analysis of the "HC10DTP" robot, with the target velocity of 250 mm/s for TCP trajectory following evaluation.

Each set of graphs in Figure 12 presents the results, reflecting the average velocity between interpolation points (top row) and trajectory deviation (bottom row), at a constant target velocity of 250 mm/s, using the second industrial robot. These results were evaluated under varying linear interpolation tolerances (from T = 0.05 mm to T = 0.5 mm). Across all three trajectories, the average velocity between interpolation points increased consistently with larger interpolation tolerances. This confirms that coarse segmentation allows the robot to maintain higher velocity trajectories. Specifically, in the first trajectory, Figure 12a, the total average velocity increased from 177.66 mm/s at T = 0.05 mm to 226.12 mm/s at T = 0.5 mm. A similar trend is observed in the second and third trajectories, Figure 12b,c, with maximum values reaching 246.53 mm/s and 238.71 mm/s, respectively. It is noteworthy that at T = 0.25 mm and above, the achieved velocities approach or exceed 240 mm/s, indicating near-full utilization of the programmed 250 mm/s velocity target.

However, smaller tolerances, such as T = 0.05 mm, consistently resulted in reduced velocities due to the denser number of interpolation points, forcing the robot controller to slow down to ensure trajectory following. It was also observed that velocities decrease significantly within a radius of 12.5 mm to 25 mm when compared to a larger radius. This indicates that the robot controller must reduce velocity in order to maintain positional accuracy when navigating sharper curves. This velocity reduction is necessary to minimize tracking errors and vibrations as robot dynamics become more difficult to control at the limits of its mechanical capabilities.

Across all three trajectories, the average velocity between interpolation points increased consistently with larger interpolation tolerances. This confirms that coarse segmentation allows the robot to maintain higher velocity trajectories. Specifically, in the first trajectory, Figure 12a, the total average velocity increased from 177.66 mm/s at T = 0.05 mm to 226.12 mm/s at T = 0.5 mm. A similar trend is observed in the second and third trajectories, Figure 12b,c, with maximum values reaching 246.53 mm/s and 238.71 mm/s, respectively. It is noteworthy that at T = 0.25 mm and above, the achieved velocities approach or exceed 240 mm/s, indicating near-full utilization of the programmed 250 mm/s velocity target. However, smaller tolerances, such as T = 0.05 mm, consistently resulted

in reduced velocities due to the denser number of interpolation points, forcing the robot controller to slow down to ensure trajectory following. It was also observed that velocities decrease significantly within a radius of 12.5 mm to 25 mm when compared to a larger radius. This indicates that the robot controller must reduce velocity in order to maintain positional accuracy when navigating sharper curves. This velocity reduction is necessary to minimize tracking errors and vibrations as the robot's dynamics become more difficult to control at the limits of its mechanical capabilities.

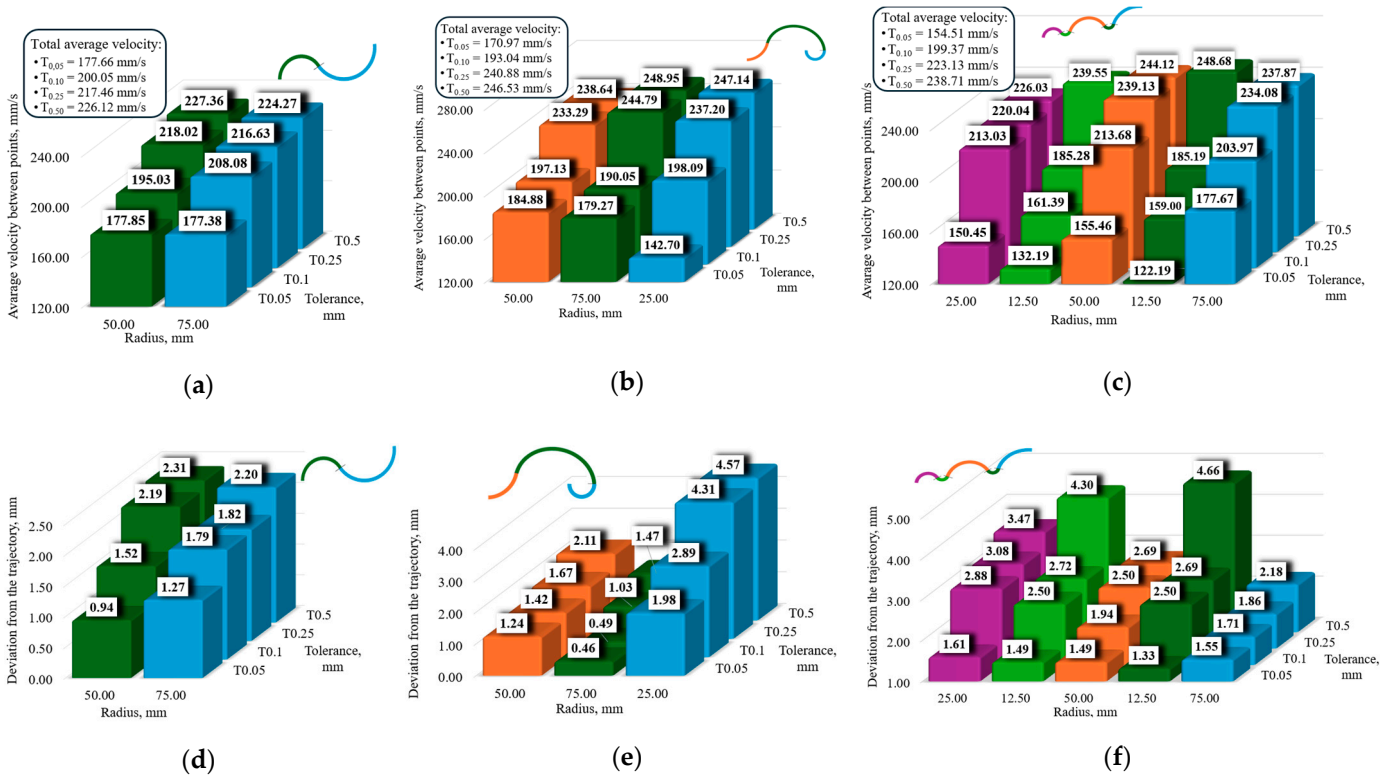


Figure 12. Average velocities and deviation from the trajectory implementing linear motion commands using "SSF2000" robot, here (**a**) average velocity values for the first trajectory; (**b**) average velocity values for the second trajectory; (**c**) average velocity values for the third trajectory; (**d**) deviations in accordance with the first trajectory; (**e**) deviations in accordance with the second trajectory; (**f**) deviations in accordance with the third trajectory.

Higher interpolation tolerances led to better velocity performance, but they also resulted in significantly increased deviations from the target trajectory. In the first trajectory, Figure 12d, deviations increased from 0.94 mm at T = 0.05 mm to over 2.3 mm at T = 0.5 mm, with a clear trend across both radiuses. The second trajectory, Figure 12e, which includes tighter turns, exhibited a dramatic increase in deviations, reaching 4.57 mm at the highest tolerance for the smallest radius (25 mm). This further highlights the sensitivity of curved trajectories to coarse linear interpolation. The third trajectory, Figure 12f, featuring multiple small-radius transitions and a curve, composed of a set of components, exhibited the most pronounced deviations. Even at T = 0.1 mm, deviation values exceeded 2.5 mm in many segments, and peaked at 4.66 mm at a tighter radius under T = 0.5 mm. This outcome confirms that trajectory complexity amplifies the effect of poor interpolation on trajectory accuracy. It is also expected that the deviations could be summed up, indicating the higher increase in deviations in the final segments.

Comparing the performance of the "SSF2000" robot with the performance of the "HC10DTP" robot, the "HC10DTP" robot is more likely to attempt to achieve the target velocity and therefore exhibits significant deviations from the trajectories. This indicates

that the "SSF2000" in this research case prioritizes speed over accuracy in its control algorithms. Nevertheless, both industrial robot setups in the research were general-purpose; the influence of different individual controller parameters was omitted. This velocity-focused approach can compromise trajectory accuracy when the robot attempts to maintain programmed speeds, particularly during complex movements. The observed deviations in trajectory indicate fundamental differences in the motion planning strategies between these robot models. In contrast, the "HC10DTP" robot is more likely to attempt to achieve the target trajectory over the target velocity, where deviations are significantly smaller compared to the second robot, but the average velocity is reduced.

In the final part of the observations, the influence of specific industrial robot motion commands was evaluated for trajectory following accuracy at varying target velocities, employing the study with the second robot model to validate the proposed methodology further. Figure 13 presents the resulting average velocity of arc segments and the corresponding trajectory deviations, which were captured using high-speed imaging during circular trajectory execution.

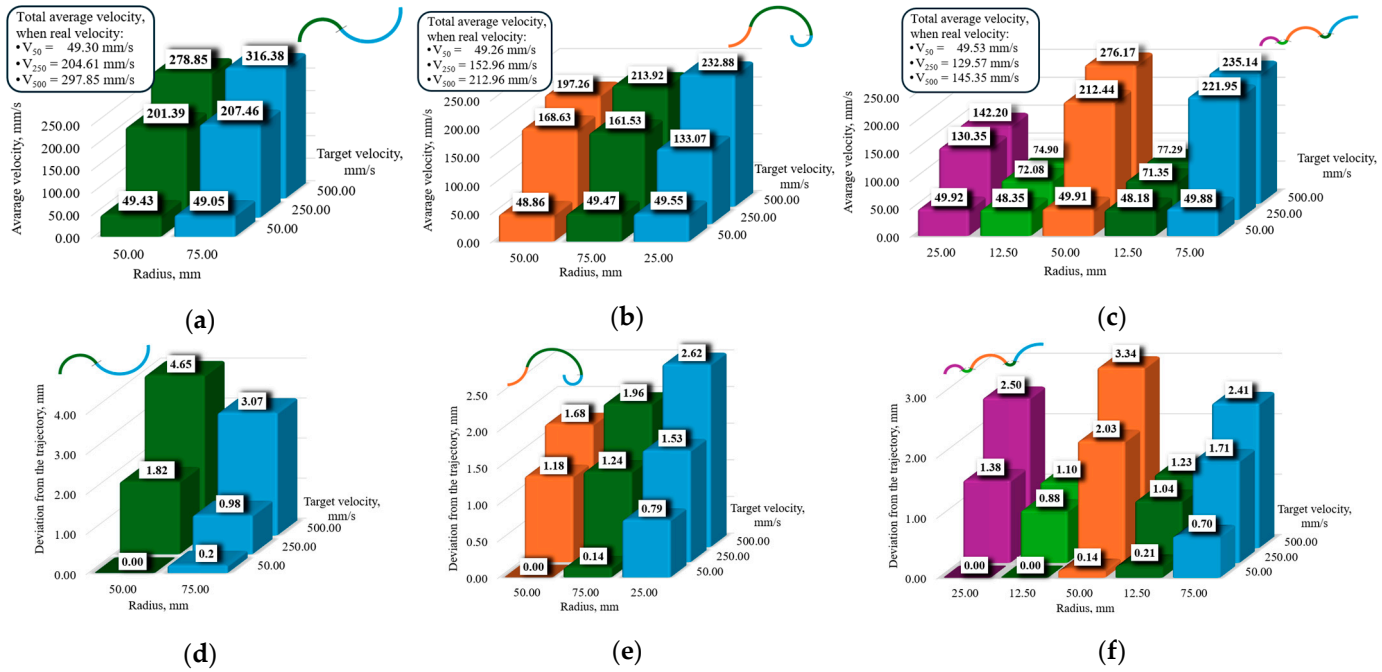


Figure 13. Average velocities and deviation from the trajectory implementing circular motion commands using "SSF2000" robot, here (**a**) average velocity values for the first trajectory; (**b**) average velocity values for the second trajectory; (**c**) average velocity values for the third trajectory; (**d**) deviations in accordance with the first trajectory; (**e**) deviations in accordance with the second trajectory; (**f**) deviations in accordance with the third trajectory.

In the first trajectory, the robot consistently achieved average velocities of 278.85 mm/s and 316.38 mm/s at 250 mm/s and 500 mm/s targets, respectively. Low-speed tracking remained consistent, with average velocities of ~49 mm/s, closely aligned with the target value. The second trajectory, characterized by tighter turns and more dynamic transitions, exhibited greater variation. While average velocities at 50 mm/s remained quite accurate (48.86–49.55 mm/s), the results at 250 mm/s and 500 mm/s decreased in tighter-radius regions, with the lowest value being 133.07 mm/s at 250 mm/s for R = 25 mm. This suggests that circular commands are more sensitive to curvature in high-speed settings. The findings indicate that the arc with a radius of 25 mm is more efficient in terms of speed, due to the greater deviation and cutting of the trajectory in a shorter way. In the third and most complex trajectory, velocity performance declined more significantly in smaller radius

segments. While the robot performed adequately at low speeds (~49 mm/s), average velocities at 250 mm/s decreased to 130.35 mm/s in the tightest curves (R = 25 mm) and even less in the radius of 12.5 mm. These findings indicate that while circular interpolation improves flow and continuity, velocity utilization is limited in areas with tight curvatures. Compared to linear interpolation at the same robotic settings and velocity—especially under T = 0.5 mm—circular commands result in marginally lower peak average velocities. However, they enable smoother motion transitions and improve trajectory behaviour.

As illustrated in Figure 13, deviations remain negligible at low speeds (close to 0 mm) and increase progressively with velocity. At a velocity of 500 mm/s, the deviation peaks at 4.65 mm for R = 50 mm, indicating that the dynamic effects become more pronounced as the velocity increases, even with circular command modes. The second trajectory demonstrates a similar trend, exhibiting higher deviations in a tighter radius and higher velocities, reaching a maximum deviation of 2.62 mm at a target velocity of 500 mm/s for R = 25 mm. It is noteworthy that at a target velocity of 50 mm/s, the deviations are negligible (<0.2 mm), demonstrating high stability under slower conditions. In the third trajectory, which encompasses the most intricate geometries, deviations are up to 2.5 mm, with the largest errors again observed at 500 mm/s and small-radius segments (R = 25 mm). Furthermore, an increase in error is observed with an increase in radius of 50 mm at a target velocity of 500 mm/s. Such deviations are evident in the first trajectory with the same radius, suggesting that the deviations are likely due to the robot's dynamic limitations when attempting to maintain constant velocity through curved trajectories. At this specific combination of radius and velocities, the controller struggles to precisely follow the programmed trajectory while balancing acceleration constraints and kinematic capabilities. This finding reinforces the conclusion that, while circular motion ensures continuity, it is not an effective solution for eliminating trajectory-following errors during high-speed execution in constrained geometries. A comparison of the results obtained from linear interpolation at T = 0.5 mm with those from circular interpolation reveals the latter to be less prone to sudden jumps in the trajectory and to maintain smaller deviations at low to medium speeds. However, at a velocity of 500 mm/s, both methods exhibit similar levels of error in tight curves, proposing that control strategy limitations, rather than interpolation type, may dominate at high-speed limits.

4. Discussion

Provided analysis of two robot types, the accuracy of the trajectory, and opened interesting findings. Firstly, different types of robots encounter varying positioning error values in their trajectories when implementing first- and second-degree interpolation (Figure 14). It is necessary to note that robots had different lifetimes and histories of operating time, but no one faced error notifications during operation, i.e., all operations during experimental research were treated as correct from the robot hardware and software. The quality of trajectory planning and local interpolation errors influence jerks and impacts, which negatively impact robot lifetime and quality. Implementing second-degree interpolation ensures a constant velocity along the trajectory regardless of the waypoint density in the curve.

The velocity of robot movement and the accuracy of waypoint passing are controversial: an increase in velocity diminishes accuracy, but different waypoint positions along the trajectory and different interpolation methods yield different error values. Results of the experimental research revealed that positioning and trajectory waypoint passing accuracy depend on the radius of curvature of the trajectory and the interpolation degree. In the case of a big curvature radius, second-order interpolation has advantages both for accuracy and trajectory speed.

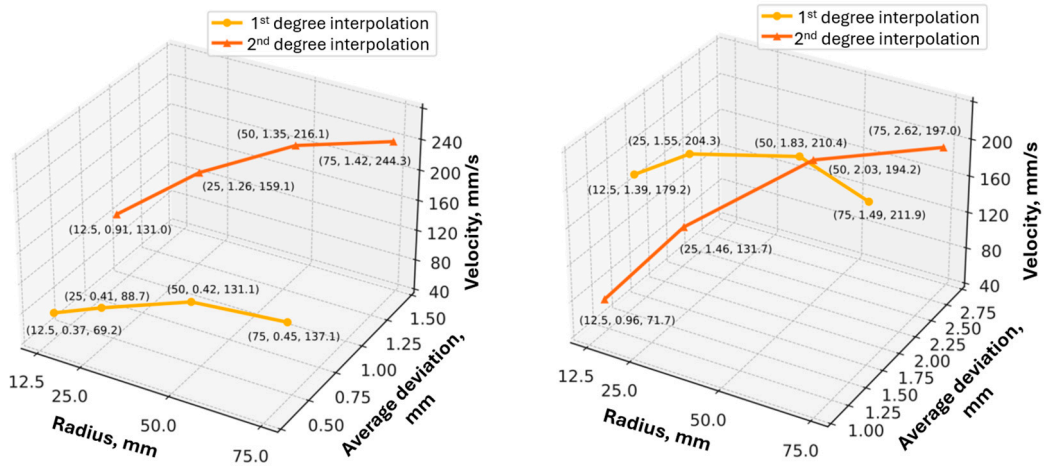


Figure 14. Summary of results with different robots: (**a**) "HC10DTP" and (**b**) "SFF2000".

The number of points typically influences the time of trajectory passing; an increase in the number of waypoints increases the passing time. This feature is especially significant in the case of interpolation of the second order. Robot positioning accuracy has the opposite tendency: an increase in the number of waypoints increases trajectory accuracy. In this case, the first-order interpolation procedure shows better accuracy than the second-order one. Additionally, the assumption about the influence of the industrial robot "HC10DTP" kinematics on the velocity downgrade is noticeable, in comparison with the "SFF2000".

The length of the trajectory and the total number of waypoints create an effect of accumulated positioning error, leading to a loss of movement accuracy. Each trajectory segment accumulates error itself, resulting in internal inconsistencies within the trajectory. The results suggest that, in the future, robots' toolpath generation algorithms should include personalization options based on the individual performance characteristics of the robot or its group, as well as real-time adaptation capabilities for individual machining or manipulation tasks. The implementation of high-speed and high-resolution cameras as optical feedback sensors in robotic setups, combined with advanced data processing solutions, represents a significant step in enhancing accuracy in robotic machining and similar operations.

The present research does not concern itself with the identification of the underlying causes of errors, which are complex and challenging to evaluate. Instead, it focuses on the evaluation and prediction of the primary error locations and the identification of methods to mitigate the impact of these errors, drawing upon the findings of the research.

5. Conclusions

This methodology facilitated a detailed evaluation of the influence of interpolation accuracy and velocity transitions on overall positioning performance, with particular relevance to high-accuracy robotic applications. The outcomes of this experimental phase provided a baseline for comparative analysis, enabling the assessment of enhanced interpolation schemes and advanced motion control techniques. The results of the performed research outline the differences between the robot trajectory's end and intermediate points, depending on the type of interpolation. Furthermore, the system's dynamic behaviour under these test conditions offered critical insights into error accumulation mechanisms and will serve as a foundation for evaluating the effectiveness of future machine learning-based compensation and adaptive trajectory generation strategies. In addition, the investigation of robot velocity fluctuations along different trajectory segments revealed important in-

formation regarding the robot controller's operation efficiency under reduced accuracy conditions. Despite the fact that these experiments were performed on a 2D plane, there are future directions to evaluate complex 3D motions by applying similar research strategies to validate them.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/machines14010008/s1: Table S1. Measurement data for the linear interpolation velocity (mm/s) of the HC10DTP industrial robot; Table S2. Measurement data for the linear interpolation deviations (mm/s) of the HC10DTP industrial robot; Table S3. Measurement data for the circular interpolation velocity (mm/s) of the HC10DTP industrial robot; Table S4. Measurement data for the circular interpolation deviations (mm/s) of the HC10DTP industrial robot; Table S5. Measurement data for the linear interpolation velocity (mm/s) of the SSF2000 industrial robot; Table S6. Measurement data for the linear interpolation deviations (mm/s) of the SSF2000 industrial robot; Table S7. Measurement data for the circular interpolation velocity (mm/s) of the SSF2000 industrial robot; Table S8. Measurement data for the circular interpolation deviations (mm/s) of the SSF2000 industrial robot.

Author Contributions: Conceptualization: M.M. and A.D.; methodology: A.D.; software: M.M. and S.P.; validation: S.P., A.D. and V.B.; formal analysis: M.M.; investigation: M.M.; resources: S.P. and A.D.; data curation: M.M. and A.D.; writing—original draft preparation: M.M.; writing—review and editing: A.D. and V.B.; visualization: M.M. and S.P.; supervision: A.D. and V.B.; project administration: A.D. and V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNC	Computer numerical control
TCP	Tool centre point
CDTW	Continuous dynamic time warping
CCD	Charge-coupled device
LiDAR	Light detection and ranging
IMU	Inertial measurement units
DH	Denavit–Hartenberg

References

1. Souflas, T.; Gerontas, C.; Bikas, H.; Stavropoulos, P. On the Optimization of Robot Machining: A Simulation-Based Process Planning Approach. *Machines* **2024**, *12*, 521. [CrossRef]
2. Lv, Z.; Wang, Y.; Zhang, G.; Wang, X. Determination method of limit interpolation points in industrial robot control system. *Robotica* **2025**, *43*, 1533–1547. [CrossRef]
3. Zu, H.; Chen, X.; Chen, Z.; Wang, Z.; Zhang, X. Positioning accuracy improvement method of industrial robot based on laser tracking measurement. *Meas. Sens.* **2021**, *18*, 100235. [CrossRef]
4. Stavropoulos, P.; Gerontas, C.; Bikas, H.; Souflas, T. Multi-Body dynamic simulation of a machining robot driven by CAM. *Procedia CIRP* **2022**, *107*, 764–769. [CrossRef]
5. Zhao, Y.; Liao, J.; Bu, S.; Hu, Z.; Hu, J.; Lu, Q.; Shang, M.; Guo, B.; Chen, G.; Zhao, Q.; et al. Automated processing and transfer of two-dimensional materials with robotics. *Nat. Chem. Eng.* **2025**, *2*, 296–308. [CrossRef]
6. He, S.; Deng, Y.; Yan, C.; Gao, Z.; Lee, C.H. A tolerance constrained robot path circular interpolation method for industrial SCARA robots. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.* **2021**, *235*, 1061–1073. [CrossRef]

7. Bilancia, P.; Ferrarini, S.; Berni, R.; Pellicciari, M. Assessing path accuracy in industrial robots via ballbar technology. *Ind. Robot*, 2024, *ahead-of-print*. [CrossRef]
8. Guo, Y.; Niu, W.; Zhou, J.; Liu, H. Near-time optimal feedrate planning for the NURBS curve considering interpolation error constraints. *Robot. Comput. Integr. Manuf.* **2024**, *86*, 102679. [CrossRef]
9. Liu, X.; Xu, Y.; Cao, J.; Liu, J.; Zhao, Y. A High-Precision Planar NURBS Interpolation System Based on Segmentation Method for Industrial Robot. *Appl. Sci.* **2023**, *13*, 13210. [CrossRef]
10. Li, R.; Ding, N.; Zhao, Y.; Liu, H. Real-time trajectory position error compensation technology of industrial robot. *Measurement* **2023**, *208*, 112418. [CrossRef]
11. Xiang, W.; Chen, J.; Li, H.; Chai, Z.; Lou, Y. Research on End-Effector Position Error Compensation of Industrial Robotic Arm Based on ECOA-BP. *Sensors* **2025**, *25*, 378. [CrossRef] [PubMed]
12. Yu, R.; Tan, X.; He, S.; Huang, Y.; Wang, L.; Peng, Y.; Wang, K. Monitoring of robot trajectory deviation based on multimodal fusion perception in WAAM process. *Measurement* **2024**, *224*, 263–2241. [CrossRef]
13. Ulrich, M.; Steger, C.; Butsch, F.; Liebe, M. Vision-guided robot calibration using photogrammetric methods. *ISPRS J. Photogramm. Remote Sens.* **2024**, *218*, 645–662. [CrossRef]
14. Lang, X.; Chen, C.; Tang, K.; Ma, Y.; Lv, J.; Liu, Y.; Zuo, X. Coco-LIC: Continuous-Time Tightly-Coupled LiDAR-Inertial-Camera Odometry Using Non-Uniform B-Spline. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7074–7081. [CrossRef]
15. Khanesar, M.A.; Karaca, A.; Yan, M.; Isa, M.; Piano, S.; Branson, D. Enhancing Positional Accuracy of Mechanically Modified Industrial Robots Using Laser Trackers. *Robotics* **2025**, *14*, 42. [CrossRef]
16. Mardanshahi, A.; Sreekumar, A.; Yang, X.; Barman, S.K.; Chronopoulos, D. Sensing Techniques for Structural Health Monitoring: A State-of-the-Art Review on Performance Criteria and New-Generation Technologies. *Sensors* **2025**, *25*, 1424. [CrossRef] [PubMed]
17. Si, L.; Wang, Z.; Li, J.; Wei, D.; Zhao, S. A novel positioning method of anti-punching drilling robot based on the fusion of multi-IMUs and visual image. *ISA Trans.* **2023**, *137*, 730–746. [CrossRef]
18. Zhang, Y.; Liang, W.; Zhang, S.; Yuan, X.; Xia, X.; Tan, J.; Pang, Z. High-precision Calibration of Camera and IMU on Manipulator for Bio-inspired Robotic System. *J. Bionic Eng.* **2022**, *19*, 299–313. [CrossRef]
19. Sharma, A.; Chaturvedi, R.; Sharma, K.; Binhowimal, S.A.; Giri, J.; Sathish, T. Enhancing weld quality of novel robotic-arm arc welding: Vision-based monitoring, real-time control seam tracking. *Ain Shams Eng. J.* **2024**, *15*, 103109. [CrossRef]
20. Huang, X.; Halwani, M.; Muthusamy, R.; Ayyad, A.; Swart, D.; Seneviratne, L.; Gan, D.; Zweiri, Y. Real-time grasping strategies using event camera. *J. Intell. Manuf.* **2022**, *33*, 593–615. [CrossRef]
21. Zhang, Y.; Wu, J.; Gao, B.; Xia, L.; Lu, C.; Wang, H.; Cao, G. Fault Types and Diagnostic Methods of Manipulator Robots: A Review. *Sensors* **2025**, *25*, 1716. [CrossRef]
22. Hou, J.; Lu, X.; Zhong, Y.; He, W.; Zhao, D.; Zhou, F. A comprehensive review of mechanical fault diagnosis methods based on convolutional neural network. *J. Vibroeng.* **2023**, *26*, 44–65. [CrossRef]
23. He, H.; Lu, C.-L.; Wen, Y.; Saunders, G.; Yang, P.; Schoonover, J.; Wason, J.; Julius, A.; Wen, J.T. High-Speed High-Accuracy Spatial Curve Tracking Using Motion Primitives in Industrial Robots. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 12289–12295. [CrossRef]
24. Le, V.; Tran, M.; Ding, S. Subtractive manufacturing of composite materials with robotic manipulators: A comprehensive review. *Int. J. Adv. Manuf. Technol.* **2024**, *135*, 81–117. [CrossRef]
25. Bilancia, P.; Schmidt, J.; Raffaeli, R.; Peruzzini, M.; Pellicciari, M. An Overview of Industrial Robots Control and Programming Approaches. *Appl. Sci.* **2023**, *13*, 2582. [CrossRef]
26. Yan, Y.; Mokri, S.S. Performance analysis of robotic arm visual servo system based on BFS-canny image edge detection algorithm. *Sci. Rep.* **2025**, *15*, 35355. [CrossRef] [PubMed]
27. Cheng, L.; Wu, J.; Guo, Y.; Li, J.; Ke, Y. An interpolation algorithm with jerk continuity for six-axis machine tools. *CIRP J. Manuf. Sci. Technol.* **2023**, *45*, 240–252. [CrossRef]
28. Qian, C.; Yao, J.; Zhang, Y. Time-Jerk Optimal Robotic Trajectory Planning Under Jerk and Continuity Constraints via Convex Optimization. *Actuators* **2025**, *14*, 272. [CrossRef]
29. Peng, P.; Wu, C.; Yang, J.; Ding, H. Tool path correction for robotic deburring using local non-rigid 3D registration. *Robot. Comput. Integr. Manuf.* **2024**, *89*, 102745. [CrossRef]
30. Robot Library | RoboDK. Available online: https://robodk.com/library (accessed on 30 April 2025).
31. Karhan, H.; Bingül, Z. Automatic Determination of the Denavit–Hartenberg Parameters for the Forward Kinematics of All Serial Robots: Novel Kinematics Toolbox. *Machines* **2025**, *13*, 944. [CrossRef]
32. Swamidoss, I.N.; Amro, A.B.; Sayadi, S. Systematic approach for thermal imaging camera calibration for machine vision applications. *Optik* **2021**, *247*, 168039. [CrossRef]

33. *ISO 9283:1998*; Manipulating Industrial Robots—Performance Criteria and Related Test Methods. ISO: Geneva, Switzerland, 1998. Available online: https://www.iso.org/standard/22244.html (accessed on 29 April 2025).
34. Marcinko, P.; Semjon, J.; Jánoš, R.; Svetlík, J.; Sukop, M.; Ondočko, Š. Analysis of the Methodology for Experimental Measuring of the Performance Criteria of the Laser-Using Collaborative Robot's Path Accuracy. *Appl. Sci.* **2024**, *14*, 1414. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.