

Article

The Effectiveness of a Robotic Workstation Simulation Implementation in the Automotive Industry Using a Closed-Form Solution of the Absolute Orientation Problem

Wojciech Andrzej Szulc  and Piotr Czop 

Faculty of Mechanical Engineering and Robotics, Department of Robotics and Mechatronics, AGH University of Science and Technology, 30-059 Cracow, Poland; szulc@agh.edu.pl

* Correspondence: pczop@agh.edu.pl

Abstract: This paper provides an in-depth analysis of a novel methodology to enhance the commissioning processes of robotic production lines in the automotive sector, with a particular emphasis on the implementation of offline programming (OLP) methods. The proposed innovative methodology, verified within the automotive industry, introduces a systematic, iterative process for calibrating and aligning the local user coordinate system (UCS) with high-precision external measurements, ensuring minimal discrepancy between simulated and actual robot paths. A significant contribution of this study is an original adjustment of the numerical algorithm applying a closed-form solution to the absolute orientation problem where unit quaternions are used to establish a UCS and evaluate positioning errors. The experimental validation study draws from 485 measurement datasets gathered across more than 300 robot stations, with each dataset comprising at least six measured point pairs, using readings from both internal robot positioning systems and a Leica AT403 laser tracker, aligned with nominal tooling values. This approach addresses discrepancies between simulated and actual environments, and our findings show an 83.51% success rate for direct implementation of simulated robot path programs. This result underscores the effectiveness of the proposed method and demonstrates the accuracy of the developed numerical algorithm, providing a reliable measure of real OLP implementation effectiveness in the automotive sector. This method further streamlines multi-robot station setup through centralized UCS alignment, significantly reducing commissioning time and enhancing efficiency in both the assembly and commissioning stages of robotized production lines. The proposed methodology facilitates precise alignment in the commissioning stage and highlights the need for synchronized simulation updates, robust offline programming practices, and regular kinematic error verification to further enhance OLP accuracy.

Keywords: base frame calibration; industrial robot; absolute orientation problem; path planning; offline robot programming



Citation: Szulc, W.A.; Czop, P. The Effectiveness of a Robotic Workstation Simulation Implementation in the Automotive Industry Using a Closed-Form Solution of the Absolute Orientation Problem. *Robotics* **2024**, *13*, 161. <https://doi.org/10.3390/robotics13110161>

Academic Editor: Andrius Dziedzickis

Received: 9 September 2024

Revised: 26 October 2024

Accepted: 28 October 2024

Published: 30 October 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robot commissioning is the process of setting up and configuring an industrial robot system to ensure it operates correctly and efficiently in a production environment. It involves several stages, from installation to the production process realized in the assumed cycle time, to verify that the robot meets the required performance and safety standards. Various offline programming (OLP) software tools have been developed for seamless planning, testing, and optimizing robot path programs in a virtual environment, instead of directly on the robot on the factory floor [1]. Industrial robotic systems, among many others, have been normalized to ensure efficient control and standardised motion planning throughout Cartesian coordinate systems [2].

The industrial robot commissioning process uses simulation programming to reduce the implementation time, which is critical in the manufacturing environment. The robot controller should be programmed the first time to achieve the highest robot positioning

accuracy, minimizing the trial-and-error approach. Tool–flange and world–base calibration involve determining the translation and rotation of the tool and base. This process allows simulated path programs to be used in real systems [3]. In many papers, this is called hand–eye calibration, where a camera is mounted on the robot’s flange, representing the tool frame origin in the equation $AX = YB$ [4–7]. Figure 1 shows the definitions of homogeneous transformations. The position of the UCS is determined as a tooling table zero position or an origin of a car reference coordinate system, the so-called car-zero [8]. The transformation X represents the TCP position in reference to the flange (Tool-0). The transformation B is internal for the robotic system, which is typically represented by a D-H notation to calculate the flange position (Tool-0) and is always provided from the robot controller [9]. Y represents world–base transformation, and A values are typically read from an external sensor, i.e., a camera, measuring arm or a laser tracker.

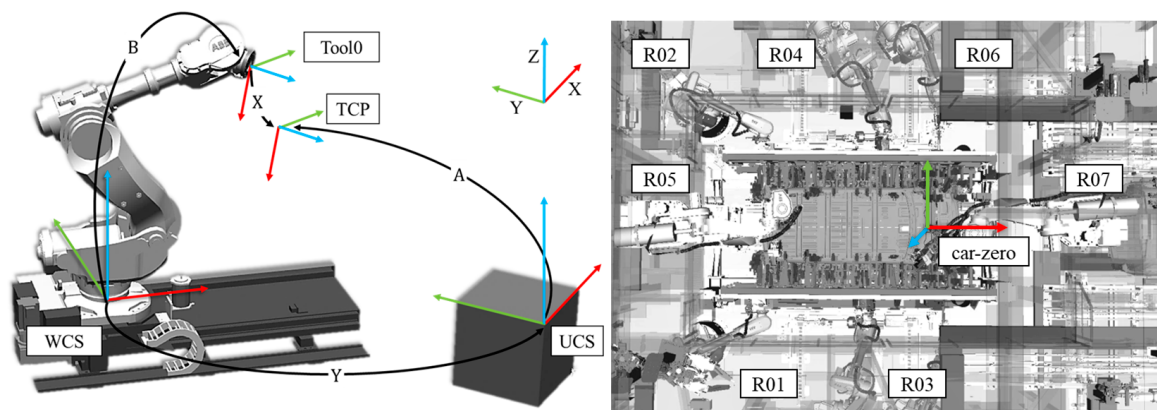


Figure 1. Coordinate systems used in industrial robot systems (WCS—world coordinate system, UCS—user coordinate system, TCP—tool center point) and an example of a car-zero reference frame in a multi-robot (R01–R07) manufacturing station.

The important aspects of effective industrial robot commissioning have been researched, developed, and patented in various papers [8,10,11]. Another advanced method in efficient production line commissioning is virtual commissioning (VC), which involves simulating and testing the entire robotic system in a virtual environment with a physical programmable logic controller (PLC) before deployment on the factory floor [12,13]. This approach allows the identification and resolution of potential issues, ensuring that the system functions as intended and reducing the need for extensive on-site adjustments.

In this paper, an efficient method of implementing simulated path programs, using an external measuring system, is presented and investigated in terms of integration effectiveness in manufacturing conditions establishing integration workflow and factors influencing the correct implementation of a robotic system.

An improper process of commissioning a robotic station increases the chances of reducing the effectiveness of the simulation implementation and increases the number of corrections required to be introduced manually in the machine executive program on-line (on the factory site). The measurement stage is a step in the robotic line commissioning process where the feasible errors can be detected, compensated, or the machine can be recalled by the manufacturer. In this research, a Leica Laser Tracker AT403 was used with its position reading level of $\pm 10 \mu\text{m}$ in distance performance, which is a class of position reading several times higher in relation to industrial robots [14]. Therefore, methods using an external measuring tool achieve better results in determining local coordinate systems than methods integrated in industrial robot systems provided by manufacturers.

The paper consists of five sections. Section 2 presents the software solution intended to be used for simulation, virtual commissioning, and measurements for collecting data, followed by the method implementing simulation into a real robotic system. At the end of the section, an industrial case example is presented. Section 3 explains a closed-form

solution of the absolute orientation problem using unit quaternions proposed by B.K.P. Horn [15]. Section 4 shows the results from 485 measurements taken in industrial conditions for ABB and KUKA robots, introducing an effectiveness parameter named the absolute average translation difference of the real system in relation to the simulated system. At the end of the section, there is a correlation verification of the collected data with the results. Section 5 summarizes and concludes the presented research.

2. Software and Methods

The automotive industry is the largest consumer of industrial robotic systems [16], especially in production processes involving the construction of car bodies, such as spot welding, arc welding, gluing, riveting, and roller hammering. These processes require the preparation of robot execution path programs that can be prepared in simulation software. There are various software tools available on the market that can be divided into solutions offered by robot producers dedicated to their products and those independent of robot type. Online and offline programming methods are commonly used in the automotive industry to reduce commissioning time, while their benefits and limitations have been reported in various papers [17,18]. Virtual commissioning (VC) is another method efficiently reducing the time required to launch a production line. It enables a configuration of industrial robots, PLCs, actuators, and I/O modules. OLP and VC are strictly connected with the digital twin (DT) concept that is part of the Industry 4.0 revolution already applied in the automotive industry, facilitating the management of a large number of industrial robots in the production process [18,19].

DT in present research in the industrial robotics field combines with virtual reality and software designed for other purposes [20], which lead to opportunities to work with industrial robotics for people not yet associated with this field. Furthermore, combining DT with automated commissioning supported by neural networks specially trained for this task [21] can accelerate the launch of robotic lines, although to commission simulated industrial robot workstations, it is necessary to compensate various inaccuracies that may occur on factory sites. These inaccuracies are related to assembly tolerances of the robot and tooling, positioning accuracy errors, and arm deflection due to tool and load weight. The manual processes of correcting inaccuracies in the production line prolong commissioning time. It is therefore important for the compensation to be applied correctly the first time to align the UCS as accurately as possible. UCS alignment can be conducted without or with external measuring tools. Types and examples of software required for simulation, virtual commissioning, and external measuring of industrial robots are presented in Table 1.

Table 1. Overview of types and examples of software for industrial robotics used for OLP, VC, and measurement.

Provider and Software	Characteristics	Sources (accessed on 1 August 2024):
Robot dedicated software		
ABB RobotStudio	Selected robot offline programming, robot environment modelling	https://new.abb.com/products/robotics/
Fanuc RoboGuide		https://www.kuka.com/
KUKA Kuka.Sim		https://www.fanuc.eu/
Staubli Robotics Suite		https://www.staubli.com/
Yaskawa MotoSim EG-VRC		https://www.motoman.com/en-us/products/robots/industrial
Robot-independent software		
<i>Siemens Process Simulate</i>	Robot offline programming, environment modeling, production planning	https://plm.sw.siemens.com/en-US/tecnomatix/products/process-simulate-software/
<i>3ds Delmia</i>		https://www.3ds.com/products/delmia
<i>RoboDK</i>		https://robodk.com/

Table 1. Cont.

Provider and Software	Characteristics	Sources (accessed on 1 August 2024):
Virtual commissioning software		
WinMod	Windows-based modeling software for DT and hardware actuators	https://www.winmod.de/
ViPer		https://www.eks-intec.de/
Measuring software		
PolyWorks Inspector	Assignment and alignment of coordinate systems to mechanical stations and collecting position data from measurements	https://www.innovmetric.com/products/polyworks-inspector
Geomagic Control X		https://oqton.com/geomagic-controlx/

Robot Positioning and Simulation Adaptation Process

The basic robot calibration is conducted at the production stage by all manufacturers. Nevertheless, optional procedures and software solutions are offered as after-sales packages in order to achieve the required process-oriented accuracy. Serial industrial robots are characterized by position repeatability of up to 0.01 mm, but their accuracy significantly differs from the repeatability value and reaches values expressed in centimetres [22,23]. There are many papers investigating this problem focusing on improving industrial robot accuracy using advanced calibration and error correction methods and addressing multiple error sources. There is in some cases a need for precise error modelling, continuous monitoring, and the use of sensors and algorithms to improve robot accuracy [22,24,25]. However, implementing constant monitoring for every production operation performed by industrial robots through external measuring systems is not feasible due to the high costs involved and the reality that many operations do not require extreme accuracy, as they are satisfied with high repeatability. When minimizing positioning errors is required (in example for laser welding, roller hamming), robot manufacturers introduce their internal solutions for improving robot positioning, the so-called Absolute Accuracy (ABB, KUKA). Despite the fact that the robot accuracy error can be reduced at the robot manufacturer site before deployment into its workplace, a large percentage of robots are still distributed without additional manufacturer calibration [26–28]. For reference, the Absolute Accuracy optional package (its name depends on the robot brand) covers the following robot positioning errors:

- Joint offsets—discrepancy between mathematical and actual zero position;
- Geometric parameters—link lengths, twist angles;
- Non-geometric parameters—joint flexibilities, base tilt;
- Environment—temperature, humidity;
- Measurement factors—non-linear encoders/resolvers.

The absence of an Absolute Accuracy optional package, the tooling and robot's large mounting tolerances to the factory floor, and the variety of robot payloads led to the development of the systematic measurement process shown in Figure 2. In this manner, the form and position error of the UCS is found. Identification of mass, center of gravity, and robot inertia should be carried out with internal robot procedures provided by robot producers, such as *LoadIdentify* (ABB), *LoadDataDetermination* (KUKA), and *PayloadIdentification* (FANUC). Methods used for load identification have been presented in [29], although the calculations used by each robot manufacturer are not publicly available.

UCS determination to a tooling station can be performed in various ways. Typically, determination of a UCS is performed with a three-point direct method covered by the robot producer's software internally prepared in the robot controller. The mathematical background of this method can be found in [30]. However, most car manufacturers follow the rule of determining each tooling UCS throughout a car zero-coordinate system and align all geometrical tooling stations to this arrangement origin. In that manner, only indirect methods can be used; for example, geometrical tooling or using a workstation with

pneumatic clamps for holding multiple parts before joining them together through welding or riveting.

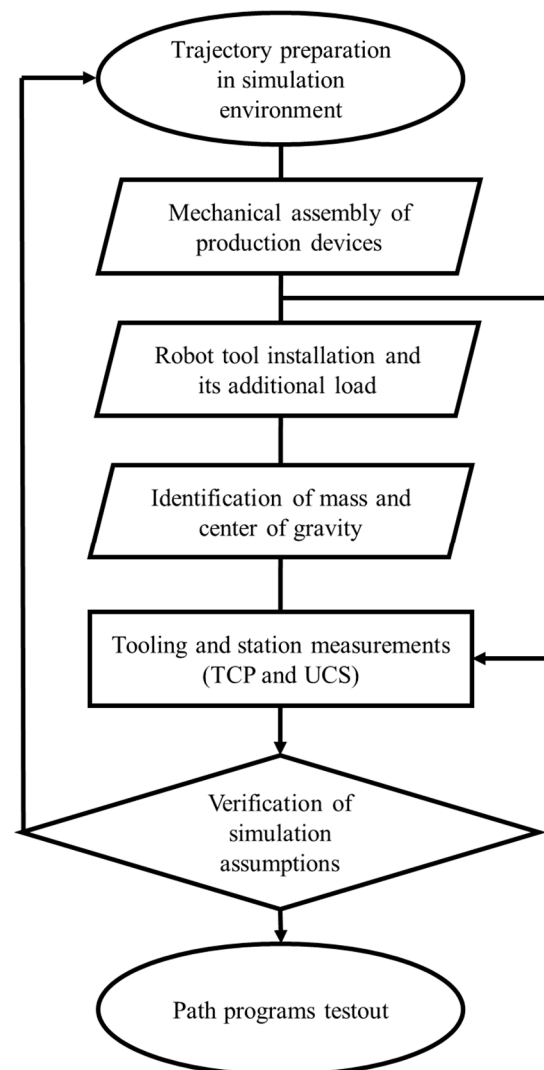


Figure 2. Block diagram of the simulation adaptation process to real systems for industrial robots.

Acknowledgement of robot absolute accuracy deficiency indicates the requirement of using an external measuring system that can take into account and determine robot error in its work field. Based on tooling nominal reference points, the UCS in the measuring system is aligned. The measuring system consists of a measuring tool, i.e., a laser tracker or measuring arm, provided on the market by producers such as FARO, Leica or API, with its measuring software for the 3D dimensional analysis to align collected point coordinates (Polyworks Inspector, Geomagic Control X). An example of a geometrical workstation and its nominal reference points used to determine a UCS is presented in Figure 3. An alignment algorithm based on the measurements is executed mostly with an iterative closest point method, which is widely implemented in various systems of machine vision and 3D reconstruction [31].

With the homogeneous equation $AX = YB$, robot position readings A in the local UCS can be carried out with an aligned measuring system for any quantity of points. For each measurement process, there are 7–10 measurement pairs obtained from the robot controller and external measurement system (laser tracker). If the residual error value for UCS position calculation in a single measured pair e_i , exceeds 1.5 mm, the point is excluded from the calculation. Such a case is treated as an error in the measurement process. If there

is a necessity to reduce measured pairs to less than 6, the entire measurement process is repeated. The equation for individual error e_i and total measurement error e is introduced in the next section, cf. (2–3). The gathered data from the robot's positioning system B and the external measurement tool is used to estimate the transformation Y with respect to the world coordinate system.

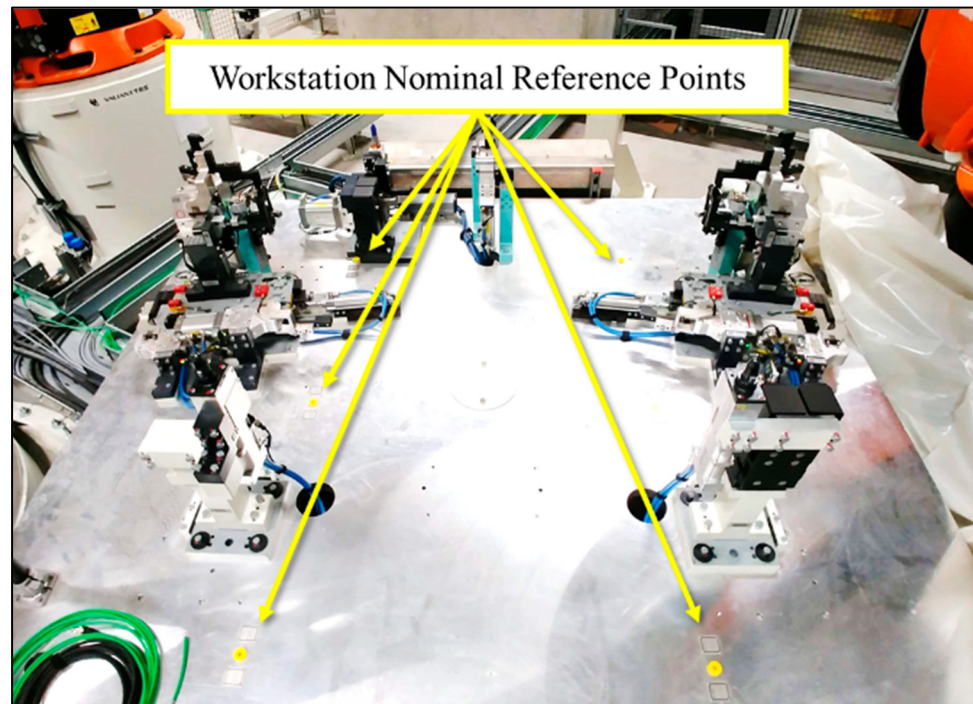


Figure 3. Example of a robot tooling workstation with nominal value references, which are used to determine the local UCS.

3. Implementation of a Closed-Form Solution

3.1. Methodology

The solution for the equation $AX = YB$ used in this paper is based on the closed-form solution of an absolute orientation problem using quaternions proposed by B.K.P. Horn [15]. The transformation scale factor is equal to 1 with no extra weighting of the given inputs. For n given sets of points, we can determine two sets of points, $\{r_{r,i}\}$ and $\{r_{l,i}\}$, representing the robot and laser respectively, where i ranges from 1 to n . Finding the translation and rotation from such sets of points consists of the steps defined in the following equations:

$$\bar{r}_0 = \bar{r}_r - R\bar{r}_l \quad (1)$$

where

$\bar{r}_0$ —Translation of robot world coordinate to UCS;

$\bar{r}_r$ —Centroid of $\{r_{r,i}\}$ robot given set of points;

$\bar{r}_l$ —Centroid of $\{r_{l,i}\}$ laser given set of points;

R —Rotation of robot world coordinate to UCS.

The centroid is a geometrical center of a set of points in three-dimensional space, which can be calculated as the arithmetic mean of the coordinates of these points. Representation of rotation varies depending on the robot manufacturer. Different forms of rotation representation and their conversions can be studied in [32]. The industrial robot systems and OLP software typically use one of the following forms of rotation representation:

- Unit quaternion q_1, q_2, q_3, q_4 — q_1 real part, q_2 – q_4 imaginary part—ABB;
- Euler angles $e = [e_z, e_y, e_x]$ —KUKA (respectively A, B, C);
- Pitch, yaw, roll—CATIA, Process Simulate, FANUC;

- 3×3 rotation matrix—calculation purposes.

For the purpose of this paper, the form of rotation matrix is used and all gathered and calculated data are unified into this form, with the final results converted back into the representation used in the selected system. The residual error equation formula for each point and average for a given set of points equals:

$$\mathbf{e}_i = \mathbf{r}_{r,i} - \mathbf{R}(\mathbf{r}_{l,i}) - \mathbf{r}_0 \quad (2)$$

$$\mathbf{e} = \frac{1}{n} \sum_{i=1}^n (\mathbf{e}_i) \quad (3)$$

In the next step, the matrix of the sum of products can be calculated:

$$\mathbf{M} = \sum_{i=1}^n \mathbf{r}'_{r,i} (\mathbf{r}'_{l,i})^T \quad (4)$$

where

$\mathbf{r}'_{r,i}$ —is a $(3 \times n)$ 3D position matrix translated by its negative centroid vector;

$\mathbf{r}'_{l,i}$ —is a $(3 \times n)$ 3D position matrix translated by its negative centroid vector.

Next, it is possible to solve the least-square problem for the rotation by defining nine individual elements of the matrix noted $S_{xx}, S_{xy}, S_{xz}, \dots, S_{zz}$, where the first index element represents the matrix row and the second is the matrix column. This form is presented in Equation (5).

$$\mathbf{M} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \quad (5)$$

Now it is possible to calculate a 4×4 N matrix of sums and differences of M matrix elements.

$$\mathbf{N} = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \quad (6)$$

The penultimate step is to find unit eigenvectors $\mathbf{q}$ and their eigenvalues of the N matrix, with four of them in this case of the 4×4 symmetric sum of product matrix. The four real eigenvalues of N, say $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, are the solutions of the characteristic polynomial (7) where $\mathbf{I}$ is an identity matrix.

$$\det(\mathbf{N} - \lambda \mathbf{I}) = 0 \quad (7)$$

Finding a most positive eigenvalue λ_m results in finding corresponding eigenvector $\overline{\mathbf{e}}_m$ of matrix N, by solving the homogenous Equation (8).

$$[\mathbf{N} - \lambda_m \mathbf{I}] \overline{\mathbf{e}}_m = 0 \quad (8)$$

$$\overline{\mathbf{e}}_m = \begin{bmatrix} r_0 \\ r_x \\ r_y \\ r_z \end{bmatrix} \quad (9)$$

Finally the searched rotation matrix is the result of expanding eigenvector $\overline{\mathbf{e}}_m$ and its elements (9) into two orthonormal matrixes 4×4 (10) and multiplication by transposition of the second matrix.

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_0 & -\mathbf{r}_x & -\mathbf{r}_y & -\mathbf{r}_z \\ \mathbf{r}_x & \mathbf{r}_0 & -\mathbf{r}_z & \mathbf{r}_y \\ \mathbf{r}_y & \mathbf{r}_z & \mathbf{r}_0 & -\mathbf{r}_x \\ \mathbf{r}_z & -\mathbf{r}_y & \mathbf{r}_x & \mathbf{r}_0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_0 & -\mathbf{r}_x & -\mathbf{r}_y & -\mathbf{r}_z \\ \mathbf{r}_x & \mathbf{r}_0 & \mathbf{r}_z & -\mathbf{r}_y \\ \mathbf{r}_y & -\mathbf{r}_z & \mathbf{r}_0 & \mathbf{r}_x \\ \mathbf{r}_z & \mathbf{r}_y & -\mathbf{r}_x & \mathbf{r}_0 \end{bmatrix}^T \quad (10)$$

After determining the rotation matrix $\mathbf{R}$, we can solve Equation (1), and find all components necessary to identify the elements needed for calculating the translation and rotation of the UCS system.

3.2. Single Measurement Case Example

In this subsection, a single measurement example is presented for the data collected from the KUKA robot system with a Leica AT 403 measuring laser tracker. Laser coordinates have been gathered by a position measuring system reflector with an offset of a robot flange, represented in Table 2 as a TCP translation. All the robot positions need to be translated by an inverse matrix of the TCP translation vector before being introduced into the absolute orientation closed-form solution presented in the previous subsection. The difference of TCP translation has to be introduced, as the reflector position represents values translated by such a vector from the robot flange (tool-0). Robot readings are values of the robot flange center position and orientation read respect to the origin of the internal world coordinate system. TCP values can be obtained directly from the robot either via the measuring system or internal robot measuring procedure ($\mathbf{X}$ in the equation $\mathbf{AX} = \mathbf{YB}$).

Table 2. Example of collected input data for a single measurement.

Name	Laser			Robot					
	X [mm]	Y [mm]	Z [mm]	X [mm]	Y [mm]	Z [mm]	A [deg]	B [deg]	C [deg]
P1	570.61	−906.16	255.38	7117.87	2873.18	1677.75	−57.930	27.843	−168.812
P2	204.88	−967.38	−610.15	7038.67	2937.71	771.54	67.407	1.981	−154.659
P3	−674.73	−535.68	889.32	6690.03	1372.91	1690.19	−153.53	5.917	154.202
P4	−1114.72	486.80	929.25	5668.70	1028.90	1644.69	48.515	−10.460	−95.022
P5	−1182.12	579.21	249.41	5651.42	1287.94	892.77	−54.790	72.622	176.770
P6	−611.1	−214.26	845.64	6344.99	1493.69	1600.83	−130.350	−9.065	89.030
P7	305.84	−631.59	261.23	6838.41	2661.65	1508.62	−76.451	73.604	142.873
P8	534.7	−990.66	−95.23	7089.16	2981.69	1403.89	76.353	−13.774	−123.964
TCP translation	---	---	---	−20.67	89.73	33.44	---	---	---

In Table 3, the results and difference between simulation assumptions and real values are presented. All Cartesian direction differences are very small; similarly, rotation does not vary from theoretical assumptions and the average residual error is far less than 1.5 mm. These resulting UCS values are to be implemented to the robot controller. A high value of the average residual error leads to verifying each single point residual error that can be removed from the calculation, which later on can be repeated without the need to repeat the whole measurement process. In this research, as mentioned previously, the minimal number of points required for measurement acceptance is 6.

Table 3. Comparison between offline assumptions (UCS simulation) and corresponding measured (UCS result) values.

	X [mm]	Y [mm]	Z [mm]	A [deg]	B [deg]	C [deg]	e [mm]
UCS Result	6125.390	2445.560	1168.990	90.529	−24.476	−0.313	0.46
UCS Simulation	6130.000	2444.711	1156.772	90.000	−25.000	0.000	---
$\Delta(X, Y, Z, A, B, C)$	4.61	−0.849	−12.218	−0.5292	−0.524	0.313	---

4. Results

In the research paper, the implementation of industrial robot simulation programs into a real-world system was evaluated using the same methodology as the single measurement case described in Section 3.1. This evaluation was conducted across 485 measurements, resulting in verification of the true effectiveness of OLP implementation. Forty measurements originated from ABB robots while the rest of the performed measurements originated from KUKA industrial robots. The results were categorized based on Formula (11), which calculated the absolute average translation difference between the real system and the simulated system.

$$\overline{\Delta T} = \left(\frac{|X_{OLP} - X_M| + |Y_{OLP} - Y_M| + |Z_{OLP} - Z_M|}{3} \right) \quad (11)$$

The maximum acceptable tolerances for implementing simulated path programs were set to ± 100 mm, considering translation movement from the simulation assumptions. The achieved effectiveness rate reached 83.51%. Complete data aggregation is presented in the Table 4. The following factors were identified as potential sources of error and should not be implemented as path programs:

1. Incorrect setup of the user coordinate system (UCS) in the simulation environment, deviating from the production plant's guidelines;
2. Position shifting of the mechanical executive systems on the production line without corresponding adjustments in the simulation environment;
3. Exceeding assembly tolerance limits for the executive systems;
4. Misdefining of the robot tools with its TCP and/or tool load values.

Table 4. The effectiveness of the robotic workstation simulation in industrial conditions, participating in the experiment.

Group	$\overline{\Delta T}$ [mm]	Amount	Percentage	e [mm]	stddev. e
1	$\overline{\Delta T} < 10$	295	60.82%	0.429	0.236
2	$10 < \overline{\Delta T} < 25$	79	16.29%	0.432	0.219
3	$25 < \overline{\Delta T} < 50$	22	4.54%	0.344	0.208
4	$50 < \overline{\Delta T} < 100$	9	1.86%	0.452	0.240
5	$\overline{\Delta T} > 100$	80	16.49%	0.461	0.272
Total success rate	---	405/485	83.51%	0.431	0.240

The results in groups 1–4 fulfilled the criteria for the program implementation from simulation environments to real systems. For the selected groups, the statistical measurement of the Pearson correlation was performed and showed a very high positive correlation between the real and simulated systems. A statistical analysis for selected implementations was performed, and Figure 4 presents a distribution of the position, dispersion, and shape of the average difference of $\overline{\Delta T}$. Pearson correlation coefficients for each direction

in groups 1–4 were $r_X = 0.99992167$, $r_Y = 0.99990738$, and $r_Z = 0.9992458$. The average measurement error across all result groups was 0.431, indicating accurate performance in all measurement activities.

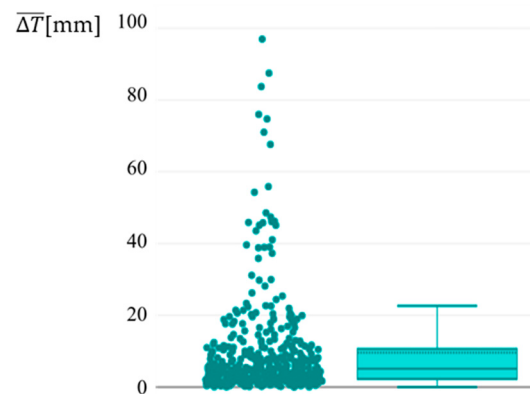


Figure 4. Distribution of the position, dispersion, and shape of the average difference of the translation of the simulated system to the real system for groups 1–4.

5. Summary and Conclusions

This paper presents a comprehensive review of modern methods designed to enhance the efficiency of commissioning processes for robotic production lines while verifying the effectiveness of offline programming (OLP) in industrial robot programming. A significant innovation introduced in this study is the use of unit quaternions to address the challenge of determining the user coordinate system (UCS) on the factory floor. The closed-form solution of absolute orientation using unit quaternions was effectively deployed across 485 measurements, each consisting of a minimum of six measured pairs, with a residual error e_i below 1.5 mm, sourced from industrial robot readings and a Leica AT403 laser tracker, aligned with nominal tooling workstation values.

The effectiveness of the commissioning process was evaluated through the absolute average translation difference between the real system and the simulated system ($\overline{\Delta T}$). The presented value of 83.51% success rate for the direct implementation of simulated robot path programs showed the true effectiveness of implementing OLP programs. The analysis revealed that the differences between OLP and the real system for 405 measurements allow use of path programs in the manufacturing line on the first attempt. In the automotive industry, which typically uses arms with lengths ranging from 2 to 4 m, a ± 100 mm margin is at the edge acceptance. In industrial settings, component assembly tolerances and safety margins of path programs prepared in OLP typically allow these kinds of adjustments to accommodate such discrepancies. In order to reduce the differences between simulation assumptions and real systems, it is necessary to:

- Synchronize simulation with real environment data to more adequately reflect factory conditions including dynamic updates in simulated environment;
- Apply robust offline programming practices with tests at significant stages of program preparation;
- Enhance robots with absolute accuracy correction matrixes and intelligent optimization approaches;
- Regularly verify the kinematic error in robotic systems, especially in systems that are using dynamic position modification.

The methodology developed for the simulation adaptation process of industrial robots ensures high-quality readings from the robot controller, facilitating measurement processes conducted in two stages of the robotic production line assembly and commissioning—both with and without a tool mounted on the robot flange. Additionally, the approach to UCS determination can significantly reduce UCS determination time, particularly in multi-robot

stations (see example in Figure 1). This is achieved throughout a single calibration of the measuring system and a one-time alignment to the workstation.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/robotics13110161/s1>, SupplementaryData_Czop-Szulc_2024.

Author Contributions: Conceptualization, W.A.S. and P.C.; methodology, W.A.S.; software, W.A.S.; validation, W.A.S. and P.C.; formal analysis, P.C.; investigation, W.A.S. and P.C.; resources, W.A.S.; data curation, W.A.S.; writing—W.A.S. and P.C.; review and editing, P.C.; visualization, W.A.S.; supervision, P.C.; project administration, W.A.S. 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 the study are included in the article/Supplementary Materials. Further inquiries can be directed to the corresponding author/s.

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

References

1. Mourtzis, D.; Doukas, M.; Bernidaki, D. Simulation in Manufacturing: Review and Challenges. *Procedia CIRP* **2014**, *25*, 213–229. [CrossRef]
2. ISO 9787:2013; Robots and Robotic Devices—Coordinate Systems and Motion Nomenclatures 2013. International Organization for Standardization: Geneva, Switzerland, 2023.
3. Zhuang Zvi Roth, H.S.; Sudhakar, R. Simultaneous Robot/World and Tool-Flange Calibration by Solving Homogeneous Transformation Equations of the Form $AX = YB$. *IEEE Trans. Robot. Autom.* **1994**, *10*, 549–554. [CrossRef]
4. Enebuse, I.; Foo, M.; Ibrahim, B.S.K.K.; Ahmed, H.; Supmak, F.; Eyobu, O.S. A Comparative Review of Hand-Eye Calibration Techniques for Vision Guided Robots. *IEEE Access* **2021**, *9*, 113143–113155. [CrossRef]
5. Wu, L.; Ren, H. Finding the Kinematic Base Frame of a Robot by Hand-Eye Calibration Using 3D Position Data. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 314–324. [CrossRef]
6. Wu, L.; Wang, J.; Qi, L.; Wu, K.; Ren, H.; Meng, M.Q.H. Simultaneous Hand-Eye, Tool-Flange, and Robot-Robot Calibration for Comanipulation by Solving the $AXB = YCZ$ Problem. *IEEE Trans. Robot.* **2016**, *32*, 413–428. [CrossRef]
7. Condurache, D.; Ciureanu, I.-A. A Novel Solution for $AX=YB$ Sensor Calibration Problem Using Dual Algebra. In Proceedings of the International Conference on Control, Decision and Information Technologies (CODIT'19), Paris, France, 23–26 April 2019.
8. Cerjaković, E.; Topčić, A.; Lovrić, S.; Herić, M. Design and Starting up of Automated Assembly Lines in Automotive Industry. In Proceedings of the 13th International Scientific Conference, Novi Sad, Serbia, 28–29 September 2018.
9. Denavit, J.; Hartenberg, R.S. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *J. Appl. Mech.* **1955**, *22*, 215–221. [CrossRef]
10. Janssen, C.; Falk, B.; Schmitt, R. Efficient Commissioning Processes for the Automotive Final Assembly. *Key Eng. Mater.* **2014**, *613*, 288–295. [CrossRef]
11. Rokossa, D. Efficient Robot Programming With-Integrated Functional Modules: Reusing Existing Knowledge in the Commissioning of Industrial Robots. In Proceedings of the IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10–13 November 2013; pp. 10–14.
12. Lechler, T.; Fischer, E.; Metzner, M.; Mayr, A.; Franke, J. Virtual Commissioning—Scientific Review and Exploratory Use Cases in Advanced Production Systems. *Procedia CIRP* **2019**, *81*, 1125–1130. [CrossRef]
13. Striffler, N.; Voigt, T. Concepts and Trends of Virtual Commissioning—A Comprehensive Review. *J. Manuf. Syst.* **2023**, *71*, 664–680. [CrossRef]
14. Hexagon AB Product Brochure: LEICA ABSOLUTE TRACKER AT403. 2018. Available online: <https://pvp-snk.ru/image/data/01Aicon/04Laser-Tracker-Systems/01Absolute-Tracker-AT403/Absolute-Tracker-AT403.pdf> (accessed on 2 August 2024).
15. Horn, B.K.P. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *J. Opt. Soc. Am. A* **1986**, *4*, 629–642. [CrossRef]
16. One Million Robots Work in Car Industry Worldwide—New Record. *Automotive Industry Is a Front Runner in Automation 2023*; International Federation of Robotics: Frankfurt, Germany, 2023; Available online: <https://ifr.org/ifr-press-releases/news/one-million-robots-work-in-car-industry-worldwide-new-record> (accessed on 30 August 2024).
17. 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]
18. Sanneman, L.; Fourie, C.; Shah, J.A.; Delft, B. The State of Industrial Robotics: Emerging Technologies, Challenges, and Key Research Directions. *arXiv* **2020**, arXiv:2010.14537. [CrossRef]
19. Maddikunta, P.K.R.; Pham, Q.V.; B, P.; Deepa, N.; Dev, K.; Gadekallu, T.R.; Ruby, R.; Liyanage, M. Industry 5.0: A Survey on Enabling Technologies and Potential Applications. *J. Ind. Inf. Integr.* **2022**, *26*, 100257. [CrossRef]

20. Garg, G.; Kuts, V.; Anbarjafari, G. Digital Twin for Fanuc Robots: Industrial Robot Programming and Simulation Using Virtual Reality. *Sustainability* **2021**, *13*, 10336. [CrossRef]
21. Tanz, L.; Daub, R. Automated Commissioning of Offline-Generated Robot Programs. *Procedia CIRP* **2022**, *107*, 810–814. [CrossRef]
22. Zhang, Y.; Qiao, G.; Song, G.; Song, A.; Wen, X. Experimental Analysis on the Effectiveness of Kinematic Error Compensation Methods for Serial Industrial Robots. *Math. Probl. Eng.* **2021**. [CrossRef]
23. Gharaaty, S.; Shu, T.; Joubair, A.; Xie, W.F.; Bonev, I.A. Online Pose Correction of an Industrial Robot Using an Optical Coordinate Measure Machine System. *Int. J. Adv. Robot. Syststems* **2018**, *15*, 1729881418787915. [CrossRef]
24. Khanesar, M.A.; Yan, M.; Isa, M.; Piano, S.; Branson, D.T. Precision Denavit–Hartenberg Parameter Calibration for Industrial Robots Using a Laser Tracker System and Intelligent Optimization Approaches. *Sensors* **2023**, *23*, 5368. [CrossRef]
25. Song, Y.; Liu, M.; Lian, B.; Qi, Y.; Wang, Y.; Wu, J.; Li, Q. Industrial Serial Robot Calibration Considering Geometric and Deformation Errors. *Robot. Comput.-Integr. Manuf.* **2022**, *76*, 102328. [CrossRef]
26. Nubiola, A.; Bonev, I.A. Absolute Calibration of an ABB IRB 1600 Robot Using a Laser Tracker. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 236–245. [CrossRef]
27. KUKA Roboter GmbH KUKA.XRob APR 2.0—Positionally Accurate Robot. Available online: <https://www.kuka.com/> (accessed on 30 August 2024).
28. ABB Inc. Absolute Accuracy—Industrial Robot Option. Available online: https://library.e.abb.com/public/931fcb281dbe7fecc1257b1300579a6a/AbsAccPR10072EN_R5.pdf (accessed on 30 August 2024).
29. Wisama, K.; Maxime, G.; Philippe, L. Identification of the Payload Inertial Parameters of Industrial Manipulators. In Proceedings of the International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007. [CrossRef]
30. Zhang, W.; Ma, X.; Cui, L.; Chen, Q. 3 Points Calibration Method of Part Coordinates for Arc Welding Robot. In *Intelligent Robotics and Applications: First International Conference, ICIRA 2008, Wuhan, China, 15–17 October 2008*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 216–224. [CrossRef]
31. Zhang, J.; Yao, Y.; Deng, B. Fast and Robust Iterative Closest Point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3450–3466. [CrossRef]
32. Kim, S.; Kim, M. Rotation Representations and Their Conversions. *IEEE Access* **2023**, *11*, 6682–6699. [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.