

## Article

# Using Full Pose Measurement for Serial Robot Calibration

Marek Franaszek \*  and Jeremy A. Marvel Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA;  
jeremy.marvel@nist.gov

\* Correspondence: marek.franaszek@nist.gov

**Abstract:** To ensure smooth robot operations, parameters of its kinematic model and a registration transformation between robot base and world coordinate frame must be determined. Both tasks require data acquired by external sensors that can measure either 3D locations or full 6D poses. We show that use of full pose measurements leads to much smaller robot orientation errors when compared with the outcome of calibration and registration procedures based on 3D data only. Robot position errors are comparable for both types of data. The conclusion is based on extensive simulations of 7 degrees of freedom robot arm and different levels of pseudo-noise perturbing both positional and rotational components of pose.

**Keywords:** robot calibration; robot remastering; calibration uncertainty; part probing; uncertainty reduction; sensor feedback



**Citation:** Franaszek, M.; Marvel, J.A. Using Full Pose Measurement for Serial Robot Calibration. *Appl. Sci.* **2022**, *12*, 3680. <https://doi.org/10.3390/app12073680>

Academic Editor: Luis Gracia and Carlos Perez-Vidal

Received: 4 March 2022

Accepted: 31 March 2022

Published: 6 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

The topic of robot calibration is well-established, yet it is still a significant factor identified by end-users as being negatively impactful for robot usability and utility [1]. Calibration is followed by registration of robot frame to world frame so the accurate encoder angles can be obtained from inverse kinematic and fed to the robot's controller. Both procedures have a profound impact on robot performance and, as pointed out in [2], "it is impossible to distinguish the end-effector error contributed either by" incorrect model parameters or by inaccurate registration transformation.

Various methods of calibrating a robot's kinematic chain have been developed (e.g., [3–5]). Many of these methods rely on intrinsic kinematic models (e.g., [6–9]), which minimize complicated, nonlinear error functions (unless only linearized error models are considered, which may exchange uncertainty for mathematical simplicity) in at least  $N$ -dimensional space, where  $N$  is the number of controllable joints in the serial kinematic chain. Calibrations based on extended modeling (i.e., beyond rigid kinematics) include compensating for thermal effects [10], and elastostatic [11] and higher order errors [12]. Likewise, examples of non-kinematics-based calibrations can be seen in [13,14]. There are also compensation techniques that can handle both kinematic and non-kinematic errors, but they require steady calculations and application of corrections during on-line operations [15–17], or dynamically selected pre-calculated, hand-eye calibrations from a table [18].

Robot calibration procedures depend on theoretical models of the mechanical system's forward kinematic. For a serial open chain robot, the Product of Exponentials (POE)—based on screw theory—is thought to be one of the most versatile models that can handle singularities in the popular Denavit–Hartenberg (DH) parameter model [19]. For robots with revolute joints only, each joint is parametrized by a three-dimensional (3D) unit vector indicating axis of rotation, and a 3D vector of any point on the axis line. Calibration procedures for such models rely on Circle Point Analysis (CPA) applied to 3D data acquired with laser tracker or other sensor: positioning the robot into a zero-reference configuration (i.e., where all joint angles are set to zero), and then rotating each joint one by one while keeping all other joints fixed at zero [20,21]. Unfortunately, POE-based models do not

explicitly include zero offsets of encoder angles. Accurate estimating of zero offsets is critical because the largest contribution to the robot positioning error (97%) comes from incorrect zero offsets [22]. Performing the zero offsets calibration in CPA causes that errors in registration transformation and in the individual offsets accumulate. This may lead to inconsistent calibration results. For some poses, the calibration process reduced robot pose error seven-fold; for others, it actually increased error twofold [23].

A desired outcome of calibration is the error reduction in full pose of robot end-effector, i.e., in its position and orientation. However, both components belong to two different spaces: position is a vector in 3D space and its components have length units, like millimeters, while orientation matrix is parameterized by three angles in degrees. This causes a fundamental scaling problem when a full pose error is minimized (as discussed in [24], ad hoc introduced scaling factors put more weight either on linear or angular part of pose error and push optimizer towards different solutions). This may become a problem in commercial applications where not only position but also orientation of end-effector is important. For example, in automated drilling, a parallelism between the spindle axis and the normal axis of the drilling plate surface should be below  $0.2^\circ$  [25]. Small orientation error  $0.05^\circ$  required for automated riveting, drilling and spot welding was demonstrated by applying online pose corrections in [26]. Automated fiber placement is another example of industrial application where the orientation of robot end-effector is important [27].

The approach that we introduce in this paper avoids the pitfall of minimization of unbalanced 6D error. First, link twists are determined in the CPA-like procedure from 3D data. Then, using full 6D poses measured by sensors, encoder zero offsets are determined in a separate minimization. The error function used in this minimization does not depend on linear DH parameters (link lengths and linear offsets) nor on the position components of noisy 6D poses acquired by sensors. Once twists and zero offsets are known, they are inserted into another error function, which depends only on position components of sensor data. The remaining linear DH parameters are determined by minimizing this second error function. For comparison, robot calibration based on only 3D sensor data is also performed. Obtained results clearly show that orientation errors of end-effector are smaller when orientation part of 6D data is used. At the same time, the position errors are comparable for both methods.

## 2. Background

The frame  $F_{TCP}$  associated with the robot's Tool Center Point (TCP) coordinate system can be expressed as a  $4 \times 4$  homogeneous transformation consisting of a  $3 \times 3$  rotation matrix  $R$  and a  $3 \times 1$  translation vector  $t$ :

$$F_{TCP} = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (1)$$

For a serial, open-chain collaborative robot arm with  $N$  revolute joints, the frame  $F_{TCP}$  in the robot's base coordinate system can be determined using a forward kinematic model:

$$F_{TCP}(\theta_k) = F_1 F_2 \dots F_N F_T \quad (2)$$

where

$$\theta_k = [\theta_{1,k}, \theta_{2,k}, \dots, \theta_{n,k}, \dots, \theta_{N,k}] \quad (3)$$

and  $\theta_{n,k}$  is the encoder angle of the  $n$ -th revolute joint for the  $k$ -th robot configuration,  $F_n$  is the homogeneous transformation associated with the  $n$ -th joint,  $n = 1, \dots, N$ , and  $F_T$  is a transformation from the robot's flange frame to the TCP. Using the nominal DH parameters, the rotation component of each  $F_n$  can be written as

$$R_{n,k}(\theta_{n,k}, \epsilon_n, \alpha_n) = R_z(\theta_{n,k} + \epsilon_n) R_x(\alpha_n) \quad (4)$$

where  $R_z$  and  $R_x$  are rotations around  $z$  and  $x$  axis, respectively. Two angular DH parameters in (4) are  $\alpha_n$  (the link twist) and  $\epsilon_n$  (the zero offset angle for the  $n$ -th encoder). The translation component of  $F_n$  can be expressed as

$$\mathbf{t}_n = \begin{bmatrix} r_n \cos(\theta_{n,k} + \epsilon_n) & r_n \sin(\theta_{n,k} + \epsilon_n) & d_n \end{bmatrix}^T, \quad (5)$$

where  $r_n$  and  $d_n$  are two linear DH parameters (link length and offset), and  $[\dots]^T$  denotes the vector transpose.

From (2) and (4) it can be seen that the rotation part  $R_{TCP}$  of the  $F_{TCP}$  frame depends only on the rotation components

$$R_{TCP}(\theta_k) = R_{1,k} R_{2,k} \dots R_{N,k} R_T. \quad (6)$$

This is a general property of serial chain manipulators with revolute joints, and is not dependent on a particular kinematic model (here, we use the DH model for illustration purposes only). In the remainder of this paper, we use the notation

$$R_k = R_{TCP}(\theta_k, \alpha, \epsilon) \quad (7)$$

where  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]$  and  $\epsilon = [\epsilon_1, \epsilon_2, \dots, \epsilon_N]$  are the vectors of the DH angular parameters. Note that the positional component  $\mathbf{t}_k$  of the  $F_{TCP}(\theta_k)$  frame depends on joint angles and all four vectors of the DH parameters:

$$\mathbf{t}_k = \mathbf{t}_{TCP}(\theta_k, \alpha, \epsilon, \mathbf{r}, \mathbf{d}). \quad (8)$$

### 3. Determination of Link Twist

To ensure that forward kinematics correctly predict the tool pose in the robot coordinate frame, the DH parameters must be determined first during the robot calibration process. Once calibrated, they remain fixed during robot on-line operations. Calibration may be performed by installing a spherically-mounted retro-reflector (SMR) at the robot's TCP, and tracking it with a laser tracker. From four vectors of DH parameters ( $\alpha, \epsilon, \mathbf{r}, \mathbf{d}$ ), the twist angles  $\alpha$  can be determined independently of other DH parameters by using 3D data acquired for CPA procedure. The twist angle  $\alpha_n$  is defined as the angle between two consecutive joint axes of rotation,  $\mathbf{u}_n$  and  $\mathbf{u}_{n+1}$  (the last twist  $\alpha_N$  is, by definition, set to zero). If  $C_{n,K}$  denotes a set of  $K$  3D points  $\mathbf{t}_k$  calculated in (8) and acquired for  $n$ -th joint in CPA procedure, then these points are distributed along an arch (section of a circle) on a plane in 3D space. Thus, for each joint  $n = 1, \dots, N$ , a unit vector  $\mathbf{c}_n$  normal to the fitted plane can be calculated. While the exact locations of  $C_{n,K}$  points depend on all DH parameters ( $\epsilon, \alpha, \mathbf{r}, \mathbf{d}$ ), vector  $\mathbf{c}_n$  is parallel to the axis of rotation  $\mathbf{u}_n$  and, therefore,  $\alpha_n$  can be determined from the scalar product of two consecutive axes,  $\alpha_n = \arccos(\mathbf{c}_n \cdot \mathbf{c}_{n+1})$ . If  $B_{n,K}$  is the set of 3D points measured by laser tracker which correspond to  $C_{n,K}$ , then a unit vector  $\mathbf{b}_n$  normal to the plane fitted to  $B_{n,K}$  can be calculated and the angle between two consecutive  $\mathbf{b}_n$  and  $\mathbf{b}_{n+1}$  is used as the estimate of  $\alpha_n$ .

To get correctly estimated twist angles  $\alpha_n$ , two important steps must be followed. First, since  $\arccos()$  is an even function, a sign of estimated angle must be equal to the sign of the default (i.e., theoretical) twist angle,  $\text{sign}(\alpha_{0,n})$ . Second, plane fitting procedure provides only a normal to the plane, its particular direction (up or down) depends on a bounding box containing the points. To remove this ambiguity, fitted normal  $\tilde{\mathbf{b}}_n$  must obey the right hand rule together with acquired 3D points  $B_{n,K}$ , which are located on a section of a circle. Thus, the estimated corrected twist angle  $\tilde{\alpha}_n$  is determined as

$$\tilde{\alpha}_n = \text{sign}(\alpha_{0,n}) \arccos(\tilde{\mathbf{b}}_n \cdot \tilde{\mathbf{b}}_{n+1}). \quad (9)$$

#### 4. Robot Calibration Based on 3D Measurements

Once the twist angles  $\tilde{\mathbf{a}}$  are estimated, they can be inserted in (4) and the remaining DH parameters  $(\epsilon, \mathbf{r}, \mathbf{d})$  can be found in traditional calibration procedure using 3D data. Given  $K$  configurations of the arm (i.e., “poses”) defined by  $\theta_k$  ( $k = 1, \dots, K$ ), the SMR is moved to  $K$  positions  $\mathbf{g}_k = [x_k, y_k, z_k]$  in 3D Cartesian space. Since  $\mathbf{g}_k$  and  $\mathbf{t}_k$  in (8) are determined in different coordinate frames, the error function  $Err_{pos}$  in the calibration process is based on relative distances between two 3D points to avoid a dependence on a registration. For convenience, the whole set of  $K$  points can be divided in two halves and then

$$Err_{pos}(\tilde{\mathbf{a}}, \epsilon, \mathbf{r}, \mathbf{d}) = \frac{1}{K/2} \sum_{k=1}^{K/2} (L_k(\tilde{\mathbf{a}}, \epsilon, \mathbf{r}, \mathbf{d}) - h_k)^2 \quad (10)$$

where

$$L_k(\tilde{\mathbf{a}}, \epsilon, \mathbf{r}, \mathbf{d}) = \|\mathbf{t}_k - \mathbf{t}_{k+K/2}\| \quad (11)$$

and  $\mathbf{t}_k, \mathbf{t}_{k+K/2}$  are determined in (8),  $\|\dots\|$  is the Euclidean norm and  $h_k = \|\mathbf{g}_k - \mathbf{g}_{k+K/2}\|$  is a distance between two points measured by laser tracker.

Thus, the fitted DH parameters  $(\tilde{\epsilon}, \tilde{\mathbf{r}}, \tilde{\mathbf{d}})$  can be estimated by minimizing  $Err_{pos}$

$$(\tilde{\epsilon}, \tilde{\mathbf{r}}, \tilde{\mathbf{d}}) = \min_{\epsilon, \mathbf{r}, \mathbf{d}} Err_{pos}(\tilde{\mathbf{a}}, \epsilon, \mathbf{r}, \mathbf{d}). \quad (12)$$

providing the vector of link twist angles  $\tilde{\mathbf{a}}$  is known. The actual dimension of search space is  $3N - 2$  since the distance between two points in (11) does not depend on  $d_1$  and  $\epsilon_1$  (the two parameters may have arbitrary values which only affect the registration transformation between robot and sensor). In the remainder of this paper, we call this procedure Method 1.

#### 5. Calibration Based on 6D Measurements

Such data were used for robot calibration using different procedures [14,28,29]. The approach we propose calculates zero offsets  $\tilde{\epsilon}$  in a separate minimization based on orientation components of 6D poses and determined earlier twist angles  $\tilde{\mathbf{a}}$ .

In the remainder of this paper, we assume that, for each robot configuration defined by  $\theta_k$ , there is a corresponding  $3 \times 3$  rotation matrix  $\mathbf{G}_k$  provided by an external sensor. Both  $\mathbf{G}_k$  and  $\mathbf{R}_k$  in (7) are determined in different coordinate frames. If  $\Omega$  denotes the rotation component of registration matrix then, for each  $k$ -th robot orientation  $\mathbf{R}_k$  in (7) and the corresponding  $\mathbf{G}_k$  measured with the external sensor, the following relation holds:

$$\mathbf{G}_k = \mathcal{N}_k \Omega \mathbf{R}_k \quad (13)$$

where  $\mathcal{N}_k$  is a small, random rotation accounting for noise in the orientation part of 6D data acquired by sensors. For a pair of orientations  $\mathbf{G}_k$  and  $\mathbf{G}_{k'}$  (where  $k' = k + K/2$ ), matrix  $\mathbf{D}_k$  can be defined as

$$\mathbf{D}_k = \mathbf{G}_k \mathbf{R}_k^{-1} \mathbf{R}_{k'} \mathbf{G}_{k'}^{-1} = \mathcal{N}_k \mathcal{N}_{k'}^{-1} \quad (14)$$

and its angle of rotation  $\psi_k \in [0^\circ \ 180^\circ]$  is calculated as

$$\psi_k = \arccos \left( \frac{1}{2} (\text{trace}(\mathbf{D}_k) - 1) \right). \quad (15)$$

Matrix  $\mathbf{D}_k$  and its angle  $\psi_k$  depend on the measured joint angle vectors  $\theta_k$  and  $\theta_{k'}$ , the twist angles  $\tilde{\mathbf{a}}$  estimated earlier, and all zero offsets  $\epsilon_n$  for  $n = 2, \dots, N$ , which can be obtained by minimizing the error function  $Err_{rot}$

$$(\tilde{\epsilon}_2, \dots, \tilde{\epsilon}_N) = \min_{\epsilon_2, \dots, \epsilon_N} Err_{rot}(\tilde{\mathbf{a}}, \epsilon_2, \dots, \epsilon_N), \quad (16)$$

where

$$Err_{rot}(\tilde{\mathbf{a}}, \epsilon_2, \dots, \epsilon_N) = \frac{1}{K/2} \sum_{k=1}^{K/2} \psi_k(\tilde{\mathbf{a}}, \epsilon_2, \dots, \epsilon_N). \quad (17)$$

Once the zero offsets  $\tilde{\epsilon}$  are estimated, they can be inserted in (12) and the linear DH parameters  $\mathbf{r}$  and  $(d_2, \dots, d_N)$  can be found by minimizing  $Err_{pos}(\tilde{\mathbf{a}}, \tilde{\epsilon}, \mathbf{r}, \mathbf{d})$  in (10). In the remainder of this paper, we call this procedure Method 2.

To show a scaling problem when both position and rotation errors are simultaneously minimized, robot calibration was attempted by minimizing the following error function:

$$Err_{full}(\tilde{\mathbf{a}}, \epsilon, \mathbf{r}, \mathbf{d}) = Err_{pos} + wErr_{rot}, \quad (18)$$

where  $Err_{pos}$  is defined in (10),  $Err_{rot}$  in (17) and positive scaling factor  $w$  ensures correct dimensionality of  $Err_{full}$ . In the remainder of this paper, we call this procedure Method 3.

## 6. Registering Robot Frame

When all robot model parameters are known, i.e., estimated  $\tilde{\epsilon}_2, \dots, \tilde{\epsilon}_N, \tilde{\mathbf{a}}, \tilde{\mathbf{r}}, \tilde{d}_2, \dots, \tilde{d}_N$  and arbitrary values are assigned to  $d_1$  and  $\epsilon_1$ , then a registration transformation (rotation  $\Omega$  and translation  $\tau$ ) between the coordinate systems of robot and laser tracker can be determined.

There are many registration techniques, one of the commonly used was developed in [30] and is based on 3D data. For calibration Method 1 described in Section 4, where only 3D data acquired by sensor are available, there is only one possible registration transformation  $(\Omega, \tau)$ . When 6D data are available, the registration transformation can be calculated in two ways. In the first (which we name Registration (1))  $\Omega_1$  is calculated using only the 3D positional parts of full poses, as in [30]. In the second (named hereafter as Registration (2)), the rotation matrix  $\Omega_2$  is calculated as the mean rotation  $\bar{\Omega}$  calculated properly [31] from the individual matrices  $G_k R_k^{-1}$  in (13). Once  $\Omega_{1,2}$  are known, the translation vectors  $\tau_{1,2}$  can be determined as

$$\tau_i = \bar{g}_s - \Omega_i \bar{t}_r, \quad i = 1, 2 \quad (19)$$

where  $\bar{t}_r$  and  $\bar{g}_s$  are the centroids of the collected 3D positions in the robot and the external sensor frame, respectively.

## 7. Simulation

All calculations were performed in Matlab. Built-in nonlinear least-square (NLS) optimizer *lsqnonlin* with default input parameters was used to minimize the error function  $Err_{pos}$  in (10),  $Err_{rot}$  in (17) and  $Err_{full}$  in (18). As a starting point for all optimizations, default DH parameters were used.

To test the proposed calibration method, a kinematic model of a 7 degrees-of-freedom (DoF) industrial robot arm KUKA LWR 4+ was used. The robot's default DH parameters  $(\epsilon_0, \alpha_0, r_0, d_0)$  are provided in Table 1 (all angular parameters are in degrees and all linear in millimeters). Ground truth (GT) parameters used in simulations were defined as a sum of the defaults and deviations, for example  $\epsilon_{GT} = \epsilon_0 + \Delta\epsilon$ . Deviations from the default DH parameters are provided in Table 2. Two sets of arbitrarily chosen deviations were used in simulations: small deviations  $(\Delta\alpha_1, \Delta r_1, \Delta d_1)$  and large deviations  $(\Delta\alpha_2, \Delta r_2, \Delta d_2)$ . GT parameters were used to generate noisy sensor data  $G_k$  from (7) and  $g_k$  from (8)

$$G_k = \mathcal{N}_k \Omega R_k(\theta_k, \alpha_{GT}, \epsilon_{GT}) \quad (20)$$

and

$$g_k = \Omega t_k(\theta_k, \alpha_{GT}, \epsilon_{GT}, r_{GT}, d_{GT}) + \tau + \zeta_k \quad (21)$$

where  $(\Omega, \tau)$  is arbitrarily selected transformation between robot and sensor frame,  $\zeta_k$  is 3D positional Gaussian noise with standard deviation  $\sigma_p$ , and  $\xi_k$  is 3D angular Gaussian noise with standard deviation  $\sigma_a$ , which was used to generate small random rotations.

$$\mathcal{N}_k = R_{ZYX}(\xi_k). \quad (22)$$

**Table 1.** Default DH parameters.

	1	2	3	4	5	6	7
$\epsilon_0$	180	90	0	0	0	0	0
$\alpha_0$	−90	−90	−90	90	90	−90	0
$r_0$	0	0	0	0	0	0	0
$d_0$	310.5	0	400	0	390	0	78

**Table 2.** Deviations of DH parameters.

	1	2	3	4	5	6	7
$\Delta\epsilon$	0	−1.4	0.68	0.24	0.54	1.37	0.85
$\Delta\alpha_1$	0.15	−0.1	0.07	−0.04	0.02	−0.06	0
$\Delta\alpha_2$	3.35	−4.1	2.7	−3.4	4.2	−3.6	0
$\Delta r_1$	0.25	0.4	0.09	0.3	0.28	0.17	0.06
$\Delta r_2$	0.8	1.3	0.65	1.4	0.86	0.38	0.55
$\Delta d_1$	0	0.06	−0.14	0.12	0.27	0.08	0.3
$\Delta d_2$	0	0.27	−1.45	0.4	1.26	0.3	0.35

In Figure 1a, examples of histograms for  $x$  component of vectors  $\xi_k$  are shown (histograms for  $y$  and  $z$  components look similar). In Figure 1b, histograms of corresponding angles of rotation  $\beta$  of small random rotations  $\mathcal{N}_k$  are plotted. Note that histograms of  $\xi_k$  are well approximated by a Gaussian distribution while non-symmetric histograms of  $\beta$  are well approximated by a Fisher–Bingham–Kent (FBK) distribution [32]. Similar histograms of angles were observed for experimental data acquired with a marker-based pose measuring system, see Figures 1 and 3 in [33].

Tool transformation  $F_T$  needed in (7) and (8) was arbitrarily chosen with the caveat that the TCP center is not located on the last axis of rotation so that 3D data acquired for CPA procedure are located on a circle.

For each  $n$ -th join,  $K_\alpha = 40$  vectors of encoder angles  $\theta_{n,k}$  were created such that their components were all zero except  $\theta_{n,k}$

$$\theta_{n,k} = \theta_{min,n} + k\delta_\theta, \quad k = 1, \dots, K_\alpha \quad (23)$$

where  $\theta_{min,n}$  and  $\delta_\theta$  were such that all  $\theta_{n,k}$  were within a valid range of  $n$ -th encoder angles. These angles were then inserted in (21) to generate 3D sensor data from which the twist angles  $\tilde{\alpha}$  were estimated as described in Section 3. In order to estimate the remaining DH parameters ( $\tilde{\epsilon}, \tilde{r}, \tilde{d}$ ) and calculate registration transformation ( $\Omega, \tau$ ), another set of  $K = 100$  joint angle vectors  $\theta_k$  was selected in such a way that corresponding poses  $F_{TCP}(\theta_k)$  in (1) were randomly scattered in the workspace that is accessible to the robot arm. In computer simulations, this is the only restriction for selection of tool poses, but additional limitations may arise in lab experiments due to a use of a line-of-sight sensor for pose acquisition.

In addition, a separate batch of  $J = 50$  joint angles  $\theta_j$  was selected for evaluation of calibration and registration procedures. These test poses were used neither in calibration nor registration. To test the performance of all three procedures, the robot kinematic model  $F_{TCP}(\theta_j)$  in (1) was used with the parameters ( $\tilde{\epsilon}, \tilde{\alpha}, \tilde{r}, \tilde{d}$ ) estimated by Method 1, 2 and 3. For Method 1, the registration transformation ( $\Omega, \tau$ ) was calculated using 3D data. For Method 2 and 3, both registrations ( $\Omega_1, \tau_1$ ) and ( $\Omega_2, \tau_2$ ) were calculated, as described in Section 6. For each tested arm configuration  $\theta_j$  and selected  $m$ -th noise levels  $(\sigma_a, \sigma_p)_m$ , the corresponding rotation  $G_j$  and position  $g_j$  were calculated in (20) and (21) to simulate noisy 6D measurements acquired by sensor. Then, the mean of  $J$  angles  $\langle \eta_j \rangle$  of rotations  $Q_j$  and the mean of  $J$  relative distances  $\langle q_j \rangle$  were calculated, where

$$Q_j(\eta_j) = G_j^{-1}\Omega R_j, \quad q_j = \|\Omega t_j + \tau - g_j\|, \quad (24)$$



and the transformation  $(\mathbf{\Omega}, \boldsymbol{\tau})$  was appropriate for each of the three calibration procedures and (for Method 2 and 3) the appropriate Registration 1 or 2. Both calculated means  $\langle \eta_j \rangle$  and  $\langle q_j \rangle$  were used as metrics to gauge a performance of tested procedures.

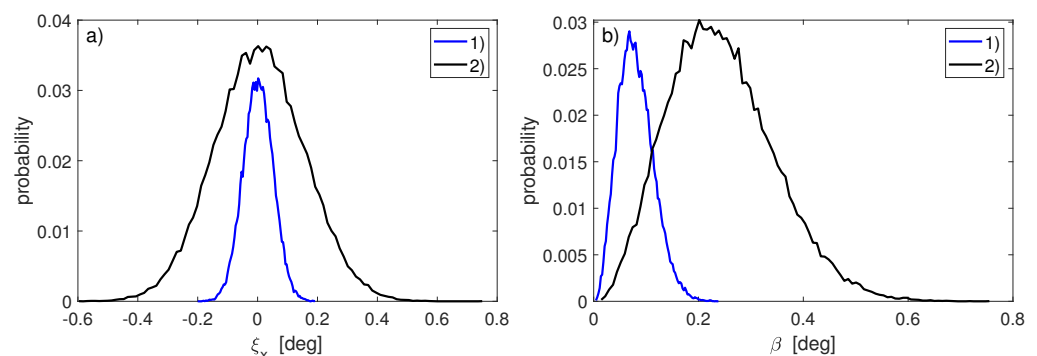
These steps were repeated for each of the selected noise level  $m = 1, \dots, M_n$  and both sets of GT parameters corresponding to two deviation vectors: small  $(\Delta \alpha_1, \Delta r_1, \Delta d_1)$  and large  $(\Delta \alpha_2, \Delta r_2, \Delta d_2)$ , as shown in Table 2.  $M_n = 16$  noise levels were equally spaced between zero and 0.15 (degrees for  $\sigma_a$  and millimeters for  $\sigma_p$ ). In order to estimate a variability of the calculated metrics, all the above calculations were repeated for  $N_{rep} = 25$  different realizations of noise (different sequences of pseudo-numbers). Thus, for each  $i$ -th instance of noise and each  $m$ -th pair of noise levels  $(\sigma_a, \sigma_p)_{m'}$ , the end-effector errors were calculated:  $v_{m,i} = \langle q_j \rangle$  for positional error and  $\rho_{m,i} = \langle \eta_j \rangle$  for angular error. As the final results, the averages and standard deviations from all repeats  $N_{rep}$  were stored for each  $m$ -th noise level:

$$\bar{\rho}_m = \frac{1}{N_{rep}} \sum_{i=1}^{N_{rep}} \rho_{m,i}, \quad \delta \rho_m^2 = \frac{1}{N_{rep}} \sum_{i=1}^{N_{rep}} (\rho_{m,i} - \bar{\rho}_m)^2 \quad (25)$$

and similarly for positional errors  $\bar{v}_m$  and  $\delta v_m^2$ .

To test a performance of the three error functions  $Err_{pos}$ ,  $Err_{rot}$  and  $Err_{full}$  used in calibration, for a few randomly selected noise repeats and strengths, minimization was restarted from 300 randomly scattered initial points (i.e., starting DH parameters) and the final optimized parameters were analyzed. In addition, for Method 3, minimization of  $Err_{full}$  was repeated for a few scaling factors  $w$  in (18).

In all simulations performed in this study, the distal variant of DH parameters was used [34]. Alternatively, the proximal variant could be used, which would affect derived from it homogeneous matrix  $F_{TCP}$ . However, not every kinematic model is suitable for describing any robot: a well-known example is a robot with two consecutive joint axes that are parallel to each other. In such a case, the DH model is not continuous and must be replaced by another model, e.g., POE [20], and parameters specific for a given model must be determined. Whichever kinematic model is selected, it is important to consistently use it in a calibration process along with other basic definitions (like use of a right-hand or left-hand coordinate system). With all procedural steps clearly defined and consistently followed, there is no ambiguity in the calibration process.



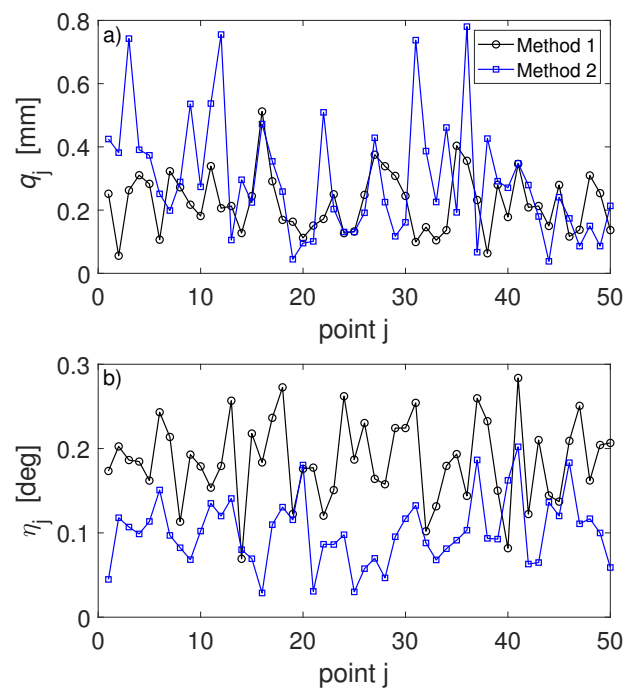
**Figure 1.** Characteristics of simulated small random rotations  $\mathcal{N}_k$  in (22): (a) histograms of  $x$  component of angle vectors  $\xi_k$ ; (b) histograms of angle of rotation  $\beta$  of rotation matrix  $\mathcal{N}_k$ . Blue lines correspond to weak noise with  $\sigma_a = 0.05^\circ$  and black lines correspond to strong noise with  $\sigma_a = 0.15^\circ$ .

## 8. Results

Fitted DH parameters revealed different amounts of variations for different simulated conditions. The twist angles  $\tilde{\alpha}$  estimated from 3D data generated for the CPA procedure showed moderate variations. The largest absolute deviation  $\delta \alpha_{max}$  from the GT value over all  $N$  joints and all simulated conditions ( $M_n$  noise levels,  $N_{rep}$  repeats and both deviations  $\Delta \alpha_{1,2}$  from the default values  $\alpha_0$ ) was  $0.3^\circ$ . Zero offsets  $\tilde{\epsilon}$  revealed larger

deviations: the largest absolute deviation  $\delta\epsilon_{max} = 1.18^\circ$ . The largest link length deviation was  $\delta d_{max} = 4.7$  mm and the largest link offset deviation was  $\delta r_{max} = 4.2$  mm. Such large differences between the fitted and the GT parameters were observed mostly for large noise levels  $\sigma_p$  and  $\sigma_a$ .

Figure 2 shows an example of robot end-effector errors at  $J = 50$  test poses. Position errors  $q_j$  and orientation errors  $\eta_j$  were calculated in (24) for robot DH parameters calibrated with Method 1 and Method 2. Presented errors were calculated for simulated sensor poses perturbed by  $i = 14$  noise realization (selected arbitrary from  $N_{rep}$  repeats) and  $m = 7$  noise levels  $(\sigma_a, \sigma_p)_m$ . These  $(q_j, \eta_j)_{m,i}$  errors were then used to calculate  $(v_{m,i}, \rho_{m,i})$  and then, mean errors  $\bar{v}_m$  and  $\bar{\rho}_m$  in (25) and the corresponding standard deviations  $\delta v_m$  and  $\delta \rho_m$  for each  $m$ -th noise level. These means and standard deviations were then used to create the plots in the remaining Figures 3–6.

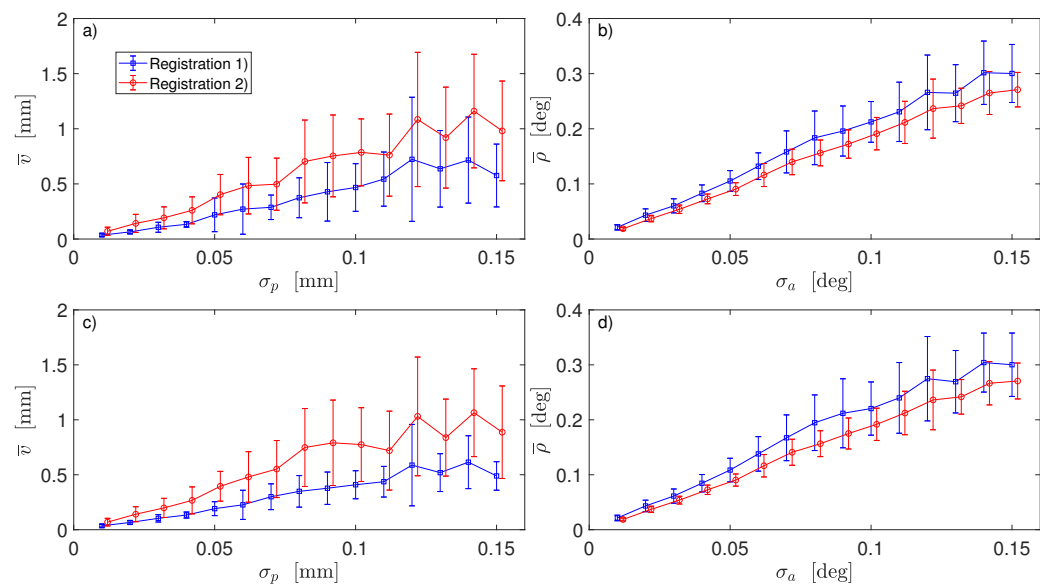


**Figure 2.** Robot end-effector errors calculated at  $J = 50$  test poses for fixed sensor noise  $\sigma_p = 0.055$  mm,  $\sigma_a = 0.055^\circ$  and one, arbitrary selected noise realization: (a) positional errors  $q_j$ ; (b) orientation errors  $\eta_j$ . Robot was calibrated with Method 1 (black lines) and Method 2 (blue lines).

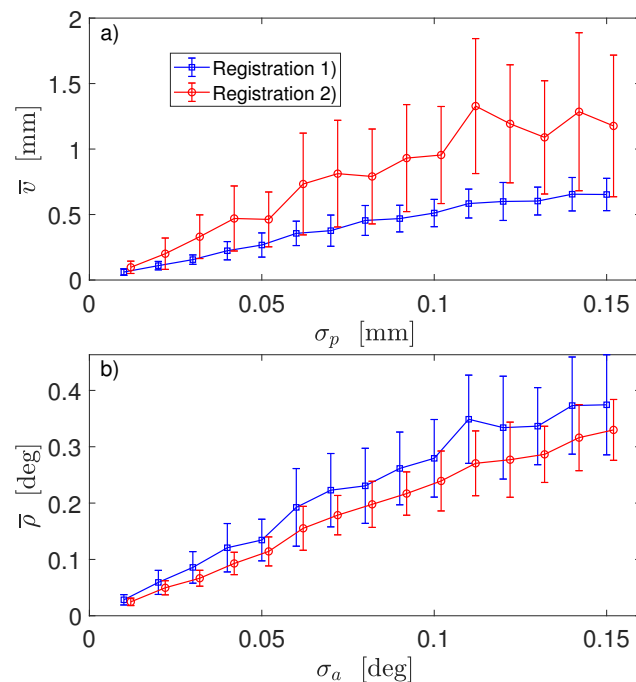
Figure 3 shows the outcomes of two registration transformations  $(\Omega_1, \tau_1)$  and  $(\Omega_2, \tau_2)$  described in Section 6. In both cases, robot was calibrated with Method 2. GT parameters used in simulation of 6D data, i.e., end-effector poses and noisy poses as measured by sensor, were obtained by modifying the default DH parameters with deviations shown in Table 2. For both registrations, mean errors were calculated at the same values of sensor noise ( $\sigma_p$  in Figure 3a,c and  $\sigma_a$  in Figure 3b,d). In each subplot, two graphs are slightly shifted horizontally only for better visualization. Error bars  $\delta v_m$  in Figure 3a,c and  $\delta \rho_m$  in Figure 3b,d are the corresponding standard deviations calculated in (25) from  $N_{rep}$  repeated simulations of noisy sensor data.

Figure 4 shows the outcomes of two registration procedures applied after robot was calibrated using Method 3 and the error function  $Err_{full}$  defined in (18) with the scaling factor  $w = 1$ . Presented results were obtained for 6D data generated with GT values of DH parameters deviating from their default values by  $(\Delta\alpha_2, \Delta r_2, \Delta d_2)$  shown in Table 2.

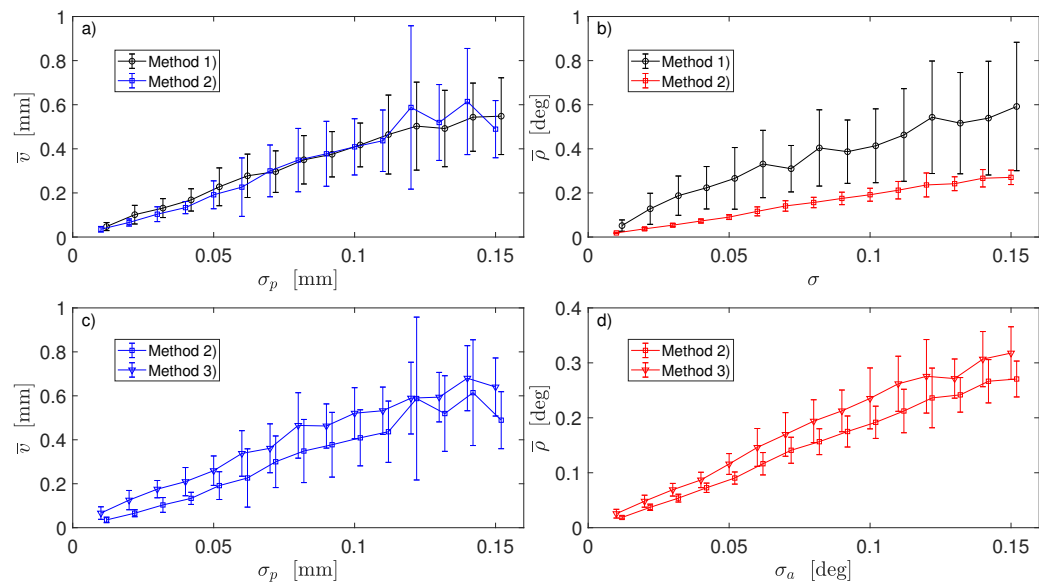




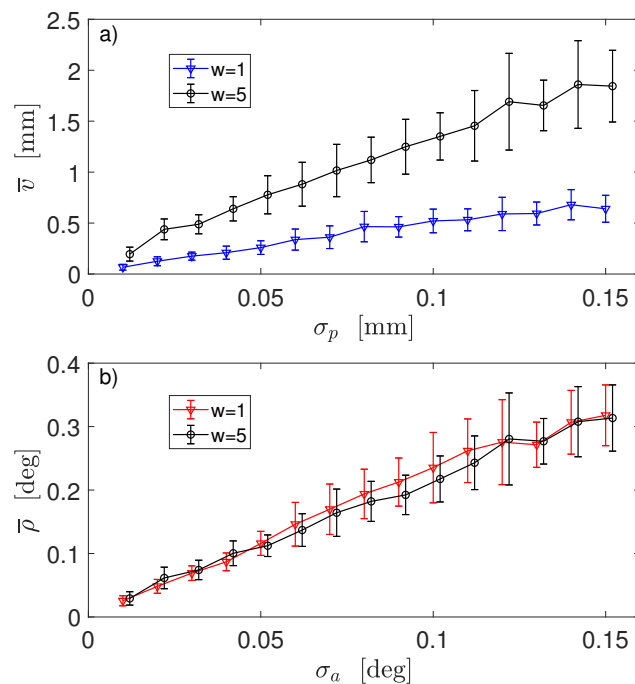
**Figure 3.** Comparison of two registration procedures for robot calibrated with Method 2 and data generated using: (a,b)—small deviations from the default DH parameters ( $\Delta\alpha_1, \Delta r_1, \Delta d_1$ ); (c,d)—large deviations ( $\Delta\alpha_2, \Delta r_2, \Delta d_2$ ). Dependence of the mean positional error  $\bar{v}$  of robot end-effector on positional noise  $\sigma_p$  in sensor 6D data in (a,d); dependence of the mean orientation error  $\bar{\rho}$  of robot end-effector on angular noise  $\sigma_a$  in sensor 6D data in (b,d).



**Figure 4.** Comparison of two registration procedures for robot calibrated with Method 3 and data generated using large deviations from the default DH parameters: (a) dependence of the mean positional error  $\bar{v}$  of robot end-effector on positional noise  $\sigma_p$  in sensor 6D data; (b) dependence of the mean orientation error  $\bar{\rho}$  of robot end-effector on angular noise  $\sigma_a$  in sensor 6D data.



**Figure 5.** Comparison of three calibration methods: (a) dependence of the mean positional error  $\bar{v}$  of robot end-effector on positional noise  $\sigma_p$ —Registration 1 was used in Method 2 (blue line); (b) dependence of the mean orientation error  $\bar{\rho}$  of robot end-effector on noise ( $\sigma = \sigma_p$  for Method 1 and  $\sigma = \sigma_a$  for Method 2)—Registration 2 was used in Method 2 (red line); (c) dependence of error  $\bar{v}$  on positional noise  $\sigma_p$ —Registration 1 was used in both methods; (d) dependence of error  $\bar{\rho}$  on noise  $\sigma_a$ —Registration 2 was used in both methods.



**Figure 6.** Comparison of robot calibrations using two different scaling factors  $w$  in error function  $Err_{full}$  in Method 3: (a) dependence of the mean positional error  $\bar{v}$  of robot end-effector on positional noise  $\sigma_p$  in sensor 6D data—Registration 1 was used; (b) dependence of the mean orientation error  $\bar{\rho}$  of robot end-effector on angular noise  $\sigma_a$  in sensor 6D data—Registration 2 was used. Data generated using large deviations from the default DH parameters.

Figure 5 shows the outcomes of three calibration procedures: Method 1 based on 3D sensor data, and Method 2 and 3 based on 6D sensor data (in Figure 5b,d noise  $\sigma = \sigma_p$  in mm for Method 1 and  $\sigma = \sigma_a$  in degrees for Method 2 and 3). Two different registration

procedures were used in robot calibration with Method 2 and 3: for positional error  $\bar{v}$ , Registration 1 was used (blue line in Figure 5a,c, the same as in Figure 3c for Method 2 and the blue line with triangle markers in Figure 5c, the same as blue line in Figure 4a for Method 3). For angular error  $\bar{\rho}$ , Registration 2 was used (red line in Figure 5b,d, the same as in Figure 3d for Method 2 and the red line with triangle markers in Figure 5d, the same as red line in Figure 4b). Error bars  $\delta v_m$  in Figure 5a,c and  $\delta \rho_m$  in Figure 5b,d are the corresponding standard deviations calculated in (25) from  $N_{rep}$  repeated simulations of noisy sensor data. On each subplot, the two graphs are slightly shifted horizontally for a visualisation effect. Robot GT parameters used in simulation of 6D data were obtained by modifying the default DH parameters with large deviations ( $\Delta \alpha_2, \Delta r_2, \Delta d_2$ ) shown in Table 2. Similar results for  $\bar{v}$  and  $\bar{\rho}$  were obtained when small deviations ( $\Delta \alpha_1, \Delta r_1, \Delta d_1$ ) were used in simulations.

Figure 6 shows outcome of robot calibration for Method 3 with two different values of the scaling factor  $w$  in  $Err_{full}$  in (18). Results for Method 3 presented in Figures 4 and 5c,d were obtained for  $w = 1$ .

For each of the selected cases where the minimization of the error function was repeated from 300 different starting points, all initial DH parameters led to the same solution. Fitted DH parameters depended on noise strengths, choice of error function and GT values of DH parameters.

## 9. Discussion

In this study, an open-chain robotic manipulator with  $N$  revolute joints was calibrated using three different methods and two different sets of data: 3D positions only, and full 6D poses. All three methods share the same strategy for determining link twists  $\tilde{\alpha}$ . Then, in Method 1, the error function  $Err_{pos}$  in (10) was minimized, and the remaining DH parameters ( $\tilde{\epsilon}, \tilde{r}, \tilde{d}$ ) were found by using 3D data only. In Method 2, a search for the zero offsets  $\tilde{\epsilon}$  was performed separately by minimizing  $Err_{rot}$  in (17), which depends only on the orientation part of full 6D data. Once the zero offsets were known, the remaining DH parameters ( $\tilde{r}, \tilde{d}$ ) were found by minimizing  $Err_{pos}(r, d)$  in (10) using only the positional part of 6D data. Such an approach reduces the dimensionality of the search space when compared with minimization of  $Err_{pos}$  in Method 1. In addition, by using angles  $\psi_k$  of relative rotations  $D_k$  in error function  $Err_{rot}$  in (17) and relative distances  $L_k$  between pairs of 3D points in error function  $Err_{pos}$  in (10), the proposed strategy decouples robot calibration from registration of the robot frame to the world frame. Different calibration strategies yielded different sets of fitted DH parameters which, in turn, led to different end-effector errors. This is expected, as the optimizer which uses different error functions and different sensor data usually converges to different solutions for the same kinematic model. It should be noted that both Methods 1 and 2 are equally valid and it is a matter of practicality which one is more useful.

In Method 2, two different approaches to registration were used. Rotation  $\Omega_1$  from the first approach minimizes distances between the sensor's 3D positions and robot's TCP points for  $K$  robot arm configurations [30]. Rotation  $\Omega_2$  is calculated as the mean rotation  $\bar{\Omega}$  of  $K$  relative rotations  $G_k R_k^{-1}$  and, thus, minimizes angular distances between orientations of TCP frame and orientations provided by sensor. Therefore, one may expect that  $\Omega_2$  is better than  $\Omega_1$  in aligning robot orientations with sensor orientations. Indeed, end-effector angular errors  $\bar{\rho}$  shown in Figure 3b,d are smaller for  $\Omega_2$  in Registration 2 (red line) than for  $\Omega_1$  in Registration 1 (blue line).

When it comes to the positional errors  $\bar{v}$ , the situation is exactly opposite. Both translation vectors  $\tau_1$  and  $\tau_2$  are calculated in (19). Since  $\Omega_2$  does not depend on positional data, the transformation  $(\Omega_2, \tau_2)$  does not minimize (in the least-square sense) the distances between sensor 3D positions and robot TCP points for  $K$  robot arm configurations. Transformation  $(\Omega_1, \tau_1)$  does minimize the distances, and therefore is expected to better align the sensor 3D positions with the robot TCP. Indeed, end-effector position errors  $\bar{v}$  shown

in Figure 3a,c are smaller for  $\Omega_1$  in Registration 1 (blue line) than for  $\Omega_2$  in Registration 2 (red line).

Analysis of the plots in Figure 3 suggests the optimal strategy: instead of choosing either the first  $(\Omega_1, \tau_1)$  or the second  $(\Omega_2, \tau_2)$  registration, take the best part from both. Use  $\Omega_2$  to transform orientations  $R_k$  of robot end-effector and use  $(\Omega_1, \tau_1)$  to transform robot TCP  $t_k$ . Outcome of such strategy is displayed in Figure 5a,b: note blue line for positional errors  $\bar{v}$  and red line for angular error  $\bar{\rho}$  indicating a use of different registrations in Method 2.

Another advantage of using  $\Omega_2$  to transform the TCP orientations rather than  $\Omega_1$  is that there is a much smaller dispersion of orientation errors  $\bar{\rho}$  for different noise realizations. The error bars in Figure 5b are much smaller for Method 2 (which leverages  $\Omega_2$ ) than for Method 1. This implies that orientations from the world coordinate system can be fed into an inverse kinematic solver more consistently and accurately.

It may appear counter intuitive that mean position errors  $\bar{v}(\sigma_p)$  and mean orientation errors  $\bar{\rho}(\sigma_a)$  calculated for the same  $m$ -th pair of noise strengths  $(\sigma_p, \sigma_a)_m$  but different GT values of DH parameters are almost the same, as Figure 3 shows. However, it should not be a surprise since we used NLS optimizer with exact error function. Scale of deviation from the default DH parameters may become an issue when the calibration is performed using approximated, linearized errors and the Jacobian is calculated at the default DH values.

Results of robot calibration obtained with Method 3 clearly reveal the consequences of scaling problem when simultaneous minimization of both position and orientation errors in one optimization is attempted, as demonstrated in Figure 6. While the mean orientation errors  $\bar{\rho}$  are almost equal for two selected values of  $w$ , the corresponding position errors  $\bar{v}$  differ substantially. This method, similarly as Method 2, uses 6D data and, therefore, two registration procedures are available. In Method 3, similarly to Method 2, smaller position errors are obtained when Registration 1 is applied to the position data and smaller orientations errors are observed when Registration 2 is applied to the orientation data, as results in Figure 4 clearly indicate. Even as both Method 2 and 3 share a possibility of using different registrations for position and rotation components of a full pose, a direct comparison between the two methods clearly points to Method 2 as a better procedure, as demonstrated by the results shown in Figure 5c,d. Thus, a use of Method 3 is discouraged.

The calibration strategy outlined in this paper was tested on a kinematic model of a serial open chain robot with revolute joints only. A question can be asked if the strategy can be applied to a more complex kinematic model when a serial chain has both revolute and prismatic joints. Acquisition of full 6D poses enables calculation of two registrations defined in (19): one of them minimizes a position error and the other minimizes an orientation error. Therefore, as long as full 6D poses are acquired, the outlined calibration strategy could in principle be used for robots with a mixture of revolute and prismatic joints. However, a presence of prismatic joints complicates the error function  $Err_{pos}$  in (10) by increasing a number of search variables and it requires further study to verify whether the strategy is beneficial also for robots with revolute and prismatic joints.

The simulation results presented in this paper raise an important, practical question about the characteristics of 6D pose measuring sensors which are used for robot calibration. Commercially available sensors allow quick acquisition of many repeated measurements, which enables the noise in recorded data to be substantially reduced by calculating mean poses. The mean position error of robot end-effector  $\bar{v}$  calculated by Method 2 is increasing with sensor position noise  $\sigma_p$ , as Figure 5a shows. If the three sigma rule is followed and approximate relation  $\bar{v} \approx 4\sigma_p$  holds, then the upper bound  $\hat{\sigma}_p$  for sensor position noise should satisfy  $12\hat{\sigma}_p < tol_p$ , where  $tol_p$  is the acceptable robot position tolerance. For orientation data, due to the strong non-symmetric FBK-like distribution of angles  $\beta$  (which accounts for deviation of noisy, instantaneous rotations from the mean rotation), the three sigma rule can be replaced by calculating quantile  $\hat{\beta}_{997}$  of angles  $\beta$  at 0.997 level. Assuming the mean orientation error of robot end-effector  $\bar{\rho}$  is four times larger than sensor's orientation noise (as shown in Figure 5b for Method 2), the upper bound for sensor

orientation noise should satisfy  $4\hat{\beta}_{997} < tol_a$  where  $tol_a$  is the acceptable robot orientation tolerance. For different robot, the dependence of end-effector error on sensor noise may be different from that shown in Figure 5a,b. Then, the estimates for upper bounds of position noise  $\hat{\sigma}_p$  and orientation noise  $\hat{\beta}_{997}$  need to be updated.

The proposed calibration strategy reduces both the position and orientation errors of the robot end-effector. Recommended procedure for serial robot calibration consists of: (1) acquiring the full 6D poses; (2) getting link twists in CPA-like procedure; (3) getting encoder zero offsets using orientation data only; (4) getting link lengths and offsets using position data only. Then, use two separate registrations to transform position and orientation component of a pose from a world to the robot frame. In summary, the dilemma of having only the position or the orientation error of the robot's end-effector minimized can be avoided and a pose with both optimized components can be fed into inverse kinematic solver.

**Author Contributions:** Conceptualization, M.F. and J.A.M.; methodology, M.F. and J.A.M.; software, M.F.; validation, M.F. and J.A.M.; formal analysis, M.F. and J.A.M.; investigation, M.F.; writing—original draft preparation, M.F.; writing—review and editing, J.A.M.; visualization, M.F.; project administration, J.A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest. Certain commercial equipment, instruments, or software are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment or software identified are necessarily the best available for the purpose.

## References

- Marvel, J.; Messina, E.; Antonishek, B.; Fronczek, L.; Wyk, K.V. *NISTIR 8093: Tools for Collaborative Robots within SME Workcells*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
- He, R.; Zhao, Y.; Yang, S.; Yang, S. Kinematic-parameter identification for serial-robot calibration based on POE formula. *IEEE Trans. Robot.* **2010**, *26*, 411–423.
- Roth, Z.; Mooring, B.; Ravani, B. An overview of robot calibration. *IEEE J. Robot. Autom.* **1987**, *3*, 377–385. [\[CrossRef\]](#)
- Hollerbach, J.M.; Wampler, C.W. The calibration index and taxonomy for robot kinematic calibration methods. *Int. J. Robot. Res.* **1996**, *15*, 573–591. [\[CrossRef\]](#)
- Elatta, A.; Gen, L.P.; Zhi, F.L.; Daoyuan, Y.; Fei, L. An overview of robot calibration. *Inf. Technol. J.* **2004**, *3*, 74–78. [\[CrossRef\]](#)
- Chen, H.; Fuhlbrigge, T.; Choi, S.; Wang, J.; Li, X. Practical industrial robot zero offset calibration. In Proceedings of the 2008 IEEE International Conference on Automation Science and Engineering, Arlington, VA, USA, 23–26 August 2008; pp. 516–521.
- 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. [\[CrossRef\]](#)
- Messay, T.; Ordóñez, R.; Marcil, E. Computationally efficient and robust kinematic calibration methodologies and their application to industrial robots. *Robot. Comput.-Integr. Manuf.* **2016**, *37*, 33–48. [\[CrossRef\]](#)
- Yang, P.; Guo, Z.; Kong, Y. Plane kinematic calibration method for industrial robot based on dynamic measurement of double ball bar. *Precis. Eng.* **2020**, *62*, 265–272. [\[CrossRef\]](#)
- Li, R.; Zhao, Y. Dynamic error compensation for industrial robot based on thermal effect model. *Measurement* **2016**, *88*, 113–120. [\[CrossRef\]](#)
- Kalas, V.J.; Vissière, A.; Company, O.; Krut, S.; Noiré, P.; Roux, T.; Pierrot, F. Application-oriented selection of poses and forces for robot elastostatic calibration. *Mech. Mach. Theory* **2021**, *159*, 104176. [\[CrossRef\]](#)
- Ma, L.M.; Fong, T.; Micire, M.J.; Kim, Y.K.; Feigh, K. Human-Robot teaming: Concepts and components for design. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 649–663.
- Joubair, A.; Bonev, I.A. Non-kinematic calibration of a six-axis serial robot using planar constraints. *Precis. Eng.* **2015**, *40*, 325–333. [\[CrossRef\]](#)
- Chen, X.; Zhang, Q.; Sun, Y. Non-kinematic calibration of industrial robots using a rigid-flexible coupling error model and a full pose measurement method. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 46–58. [\[CrossRef\]](#)
- Yang, J.; Wang, D.; Fan, B.; Dong, D.; Zhou, W. Online absolute pose compensation and steering control of industrial robot based on six degrees of freedom laser measurement. *Opt. Eng.* **2017**, *56*, 034111. [\[CrossRef\]](#)
- Zeng, Y.; Tian, W.; Li, D.; He, X.; Liao, W. An error-similarity-based robot positional accuracy improvement method for a robotic drilling and riveting system. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 2745–2755. [\[CrossRef\]](#)

17. Chen, D.; Yuan, P.; Wang, T.; Ying, C.; Tang, H. A compensation method based on error similarity and error correlation to enhance the position accuracy of an aviation drilling robot. *Meas. Sci. Technol.* **2018**, *29*, 085011. [\[CrossRef\]](#)
18. Franaszek, M.; Cheok, G.S. Using locally adjustable hand-eye calibrations to reduce robot localization error. *SN Appl. Sci.* **2020**, *2*, 839. [\[CrossRef\]](#)
19. Park, F.C. Computational aspects of the product-of-exponentials formula for robot kinematics. *IEEE Trans. Autom. Control* **1994**, *39*, 643–647. [\[CrossRef\]](#)
20. Wang, H.; Shen, S.; Lu, X. A screw axis identification method for serial robot calibration based on the POE model. *Ind. Robot Int. J.* **2012**, *39*, 146–153. [\[CrossRef\]](#)
21. Santolaria, J.; Conte, J.; Ginés, M. Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector. *Int. J. Adv. Manuf. Technol.* **2013**, *66*, 2087–2106. [\[CrossRef\]](#)
22. Conrad, K.L.; Shiakolas, P.S.; Yih, T. Robotic calibration issues: Accuracy, repeatability and calibration. In Proceedings of the 8th Mediterranean Conference on Control and Automation (MED2000), Rio, Patras, Greece, 17–19 July 2000; Volume 1719.
23. Choi, Y.; Cheong, J.; Kyung, J.H.; Do, H.M. Zero-offset calibration using a screw theory. In Proceedings of the 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Xi'an, China, 19–22 August 2016; pp. 526–528.
24. Wu, Y.; Klimchik, A.; Caro, S.; Furet, B.; Pashkevich, A. Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments. *Robot. Comput.-Integr. Manuf.* **2015**, *35*, 151–168. [\[CrossRef\]](#)
25. Posada, D.J.; Schneider, U.; Pidan, S.; Geravand, M.; Stelzer, P.; Verl, A. High Accurate Robotic Drilling with External Sensor and Compliance Model-Based Compensation. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 3901–3907.
26. 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. Syst.* **2018**, *4*, 1–16. [\[CrossRef\]](#)
27. Rousseau, G.; Wehbe, R.; Halbritter, J.; Harik, R. Automated Fiber Placement Path Planning: A state-of-the-art review. *Comput.-Aided Des. Appl.* **2019**, *16*, 172–203. [\[CrossRef\]](#)
28. Nguyen, H.N.; Zhou, J.; Kang, H.J. A new full pose measurement method for robot calibration. *Sensors* **2013**, *13*, 9132–9147. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Nubiola, A.; Bonev, I.A. Absolute robot calibration with a single telescoping ballbar. *Precis. Eng.* **2014**, *38*, 472–480. [\[CrossRef\]](#)
30. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 698–700. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Moakher, M. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.* **2002**, *24*, 1–16. [\[CrossRef\]](#)
32. Kent, J.T. The Fisher-Bingham Distribution on the Sphere. *J. R. Stat. Soc.* **1982**, *44*, 71–80. [\[CrossRef\]](#)
33. Franaszek, M.; Shah, M.; Cheok, G.S.; Saidi, K.S. The axes of random infinitesimal rotations and the propagation of orientation uncertainty. *Measurement* **2015**, *72*, 68–76. [\[CrossRef\]](#)
34. Lipkin, H. A note on Denavit-Hartenberg notation in robotics. In Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Long Beach, CA, USA, 24–28 September 2005; pp. 1–6.