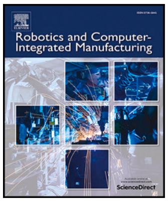
Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcm



Full length article

A novel vision-based calibration framework for industrial robotic manipulators

Hamid Majidi Balanji [a,*], Ali Emre Turgut [a], Lutfi Taner Tunc [b,c]

[a] Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey
[b] Integrated Manufacturing Research and Application Centre, Sabanci University, Istanbul, Turkey
[c] Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Industrial robot calibration
POE
Computer vision
Fiducial marker system
Vision-based robot calibration

ABSTRACT

With the increasing involvement of industrial robots in manufacturing processes, the demand for high quality robots has increased considerably. A high-quality robot is a robot having good repeatability and accuracy. Industrial robots are known to have very good repeatability, however it is not the same with accuracy. Due to harsh working conditions, accuracy of robots deteriorate over time. Calibration is a practical approach to sustain accuracy. In calibration, position and orientation of the tool center point (TCP) of a robot arm should be corrected using a tracking device with higher accuracy. Different devices such as laser-trackers, optical CMMs, and stereo cameras have been used in the literature. In this paper, a novel calibration framework is proposed based on a single camera and computer vision techniques using ArUco markers. The product of exponentials method is used for kinematic modeling of the robot to avoid the singularity. The performance of the framework is tested using computer-based simulations and using a six degree of freedom (6-DOF) UR5 robotic manipulator. Position and orientation errors are used as metrics in the experiments. The position and orientation errors in real world experiments reached to 2.5 mm and 0.2°, respectively. The result shows that the method is usable in real world scenarios.

1. Introduction

Robotic manipulators have been used in material handling, welding and painting tasks since decades [1]. In such operations, performance expectations are not very high. However, together with increasing use of industrial robots for machining applications in aerospace, automotive and power generation industries [2], performance expectations of industrial robots have increased tremendously. Performance expectations are defined in terms of accuracy and repeatability of position and orientation of the end-effector or the tool center point (TCP) of an industrial robot. Accuracy is defined as the deviation between a commanded pose and the mean of the attained poses by the TCP, and repeatability is defined as the variance of the attained poses after n repetitions [3]. Industrial robots are known to have good repeatability [4,5], however they lose their accuracy over time due to joint and gear compliance, and gear backlash [6]. Therefore, periodic calibration is needed to keep accuracy of robotic manipulators at acceptable levels [7].

Calibration is defined as: "The comparison of measurement values delivered by a device under test with those of a calibration standard of known accuracy" [8]. In the literature, there are different calibration methods [9] using different pose tracking systems; such as theodolites, coordinate measurement machines (CMMs), and laser tracker systems (LTSs). Recently, with the increase of the processing power of computers and availability of high-quality cameras, computer vision-based calibration methods have been used.

Theodolites [10] were the first devices used for TCP pose estimation in robot calibrations [11,12]. They were also used in TCP pose estimation with a vision system (VBAT) [12]. Veitschegger and Wu [13] used a CMM for calibrating a PUMA 560 robot. The results showed that the positional accuracy of their proposed measurement system was adequate for calibration. Mooring and Padavala [14] also used a CMM to calibrate a PUMA 560 robot. Due to the restriction in the size of their CMM, they were not able to calibrate the robot in its entire workspace. Nubiola et al. [4] calibrated an ABB IRB 120 robot using an optical CMM and a FARO laser tracker. They demonstrated the positional accuracy of the optical CMM was approximately equal to that of the laser tracker. Moreover, they also stated that optical CMM was cheaper and faster compared to the laser tracker. Gharaaty et al. [15] proposed an online dynamic pose correction system using an optical

[*] Corresponding author.
E-mail addresses: balanji.hamid@metu.edu.tr (H.M. Balanji), aturgut@metu.edu.tr (A.E. Turgut), ttunc@sabanciuniv.edu (L.T. Tunc).

https://doi.org/10.1016/j.rcim.2021.102248
Received 22 February 2021; Received in revised form 16 August 2021; Accepted 16 August 2021
Available online 6 September 2021

CMM to enhance the pose accuracy of a 6-DOF industrial robot that was used for riveting, drilling, and spot welding.

Laser trackers are considered as one of the most practical and reliable systems for TCP pose estimation [16], and many researchers have adopted it for robot calibration [17]. Alici and Shirinzadeh [18] developed a method to estimate the systematic positional errors of a Motoman SK 120 robotic manipulator using an LTS. Since, in their application, LTS was not able to measure the orientation of the TCP, the proposed method was limited to the estimation of position errors only. Ye et al. [19] calibrated an ABB IRB 2400/L robot with a FARO laser tracker. They used modified D-H convention for the kinematic modeling of the robot in their calibration method and improved position accuracy of the robot considerably. Nubiola and Bonev [16] improved the positional accuracy of an ABB IRB 1600 industrial robot using a 29-parameter calibration model. They used a FARO laser tracker to measure the kinematic parameters of the robot. They validated their calibration method using 1000 random configurations and this calibration method became a reference benchmark for industrial robot calibrations.

As discussed so far, several methods have been employed for pose estimation in robot calibration. Theodolite-based methods are simple, accurate, but very slow due to discrete and manual pose measurements. CMM-based methods are very accurate and fast, especially the optical-probe type but have limited use due to their small workspace [20]. LTS-based methods are the most advanced ones being accurate and popular in industry, but they are very expensive, the calibration time is too long, and their performance is degraded due to harsh working conditions in factories [21,22]. Nowadays, with the increase in the capabilities of the cameras and the processing power of computers, computer vision systems started to be used in robot calibration operations.

In computer vision-based applications, two different camera configurations are employed as shown in Fig. 1. The first one is the eye-to-hand configuration in which the camera is fixed on a stationary reference frame in the workspace of the robot. The second one is the eye-in-hand configuration where the camera is fixed on the robot. Single or multiple cameras referred to as monocular vision or stereo vision are used at each robot-camera configuration. Recently, stereo-vision techniques, which employ two cameras to extract three-dimensional position and orientation information are adopted for TCP pose estimation in industrial robot calibrations. These methods are fast, versatile, and moderately accurate [23]. Švaco et al. [24] designed a customized stereo vision system using two perpendicular CCD cameras equipped with macro lenses to improve the positional accuracy of a KUKA KR6-R900 robot. Though the calibration results were satisfactory, the manual adjustment of different robot poses was time consuming. Moller et al. [25] adopted a stereo vision system to increase the positional accuracy of a robotic manipulator used for milling. The experiments revealed that the pose information provided by the stereo vision system was more precise than the one measured by the robot. Zhang et al. [26] used a single fiducial marker and a stereo camera system for the calibration of a UR5 robot. The marker was installed on the end-effector of the robot and it was detected by the stereo camera system in the eye-to-hand configuration. The experiments and simulations showed that the method was effective and robust for robot calibration. Despite being an automated calibration method, target configuration space was limited due to FOV of the camera.

Filion et al. [27] proposed a robot calibration method using a portable photogrammetry system. The modified Denavit–Hartenberg method (modified D-H) was used for the kinematic modeling of the robot. The portable photogrammetry system needs some additional reference frames with respect to the robot and its surrounding. The system can only measure passive retro-reflective targets that are fixed on the robot. The performance of their proposed calibration method was evaluated using a 6-DOF FANUC LR Mate 200iC robot. A FARO laser tracker was employed for verification purposes. The calibration process was implemented in MATLAB. In order to avoid the singularity in the different poses of the robot, inverse kinematic solutions were employed, making the calibration method computationally expensive and slow for online applications. Although the application of portable photogrammetry tools is uncommon in industrial robot calibrations, the results showed that they could be feasible in the near future.

The robot calibration methods based on the monocular vision systems are computationally fast and cost-effective [28]. However, they are challenging methods since it is hard to compute the 3D coordinate of a point in space using a 2D projected image. There are a few studies about industrial robot calibration using monocular vision. Yong et al. [29] designed a monocular vision system for the pose determination of a Scorbot robot. For pose estimation, image processing techniques, including: (i) thresholding, (ii) noise elimination, (iii) region filling, and (iv) morphological dilation, were used in their method. The estimated pose was then fed into the D-H kinematic model of the robot. In their method, the effective camera resolution was 3664 × 2748, and camera frame rates were about 7.5 fps. Image processing and kinematic calculations were time-consuming and the camera should be installed such that its optical axis was parallel to the y-axis of the robot manipulator. Motta et al. [30] also proposed a robot calibration method based on monocular vision. Their system was composed of a CCD camera installed on the end-effector of the robot, forming an eye-in-hand configuration. They mainly concentrated on calculating a more precise camera calibration matrix, including the camera extrinsic and intrinsic parameters, by eliminating the radial distortions of digital images, which caused a considerable improvement in pose estimation accuracy. For this purpose, they used the radial alignment constraint model for camera calibration. A CMM, with the camera mounted on its table, was used to evaluate the accuracy of their proposed calibration method. The evaluation process was complicated and insufficient because the CMM was not able to measure rotation; therefore, only positional accuracy was evaluated. The proposed method was applied for the calibration of the IRB-2400 and PUMA-500 robots in real-world experiments. Du and Zhang [31] proposed an automatic vision-based calibration method using a single camera in the eye-in-hand configuration using a calibration chessboard. Generally, vision-based pose estimation methods using the calibration chessboard and a monocular vision system are inaccurate due to the noise and non-linearity of the camera lens. Thus, they used extra image processing steps to increase the accuracy of the estimated pose. Their proposed method was verified using several pick-and-place and peg-into-hole tests by using a GOOGOL GRB3016 robot. When high-resolution images were used, the pose accuracy and camera calibration errors were relatively high. When low frame rates were used to increase the pose accuracy, accuracy increased but the calibration time also increased. Moreover, singularity problems were observed since they used the standard Denavit–Hartenberg method (D-H) for the kinematic modeling of the robot.

Wang et al. [32] proposed a vision-based closed-loop calibration method based on point and distance constraints. They used the modified D-H method for kinematic modeling and inverse kinematic to solve the singularity problem. The pose of the robot was measured using a single high-resolution camera to detect a spherical ceramic ball with a diameter of 25.4 mm. The ball was installed at the center of the FOV of the camera to reduce the distortion effect of the radial lens. The ball center was taken as a constraint point, and the camera position was set such that its optical axis passed through the center of the ball. Although their method improved the positional accuracy of the Stäubli TX-60 robot and was easy to use, image processing at a resolution of 3384 × 2704 decreased the processing speed considerably and required high computation time. Moreover, it was difficult to keep the center of the ball and camera optical axis coincident, making it hard to lower the positional error.

All sensors, including cameras have some degree of uncertainty due to environmental conditions and inherent process noise [33]. Due to

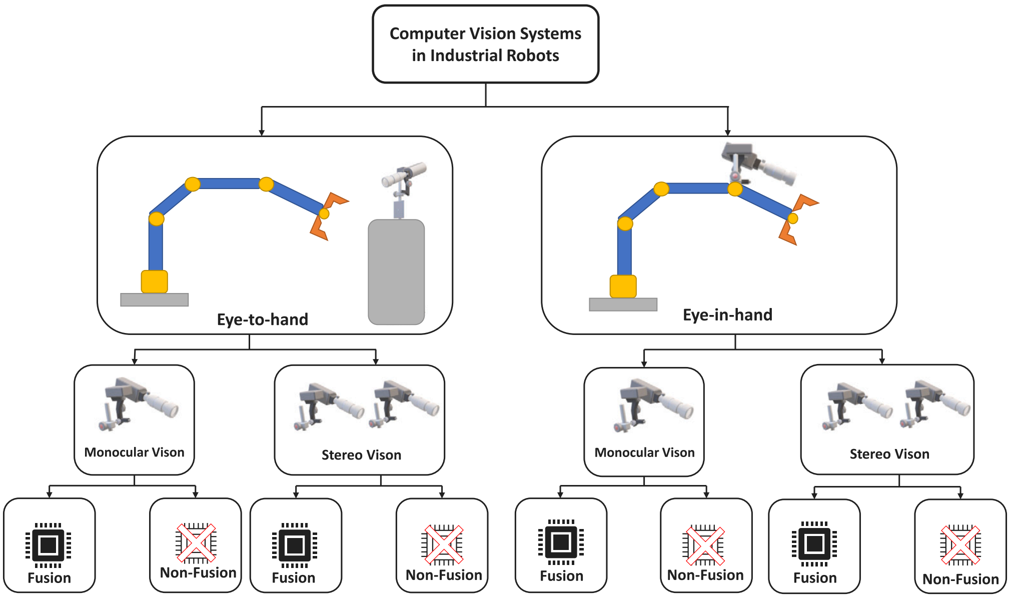


Fig. 1. Application of computer vision systems in industrial robots.

these issues, in vision-based systems, other sensors like inertial measurement units (IMUs) [31,34] were used to increase the accuracy of measurements. Liu et al. [34] proposed a pose estimation algorithm for robot calibration using a multi-sensor fusion approach. They fused the position data obtained from a photogrammetry system and orientation data obtained from a digital inclinometer using a Kalman filter and an optimal fusion algorithm to estimate the pose of the end-effector of the robot. The results showed that the positional errors decreased to less than 1 mm. Although the sensor fusion-based techniques have high accuracy, they were not employed frequently in real world scenarios due to their complexity and high computational requirements. Du et al. [35] proposed another fusion-based automatic robot calibration method using Kalman filter. The standard D-H method was used for the kinematic modeling of the robot during the calibration procedure. The position and orientation of the end-effector of the robot were measured using a position sensor and an IMU attached to the end-effector of the robot. An Extended Kalman Filter (EKF) was used to fuse the position and orientation data to obtain the 6-DOF pose of the end-effector of the robot. The temperature drift of the IMU due to the change in the environment of the experiments rendered the method impractical, and the use of inverse kinematic methods to avoid the singularity problem increased the computational requirements of the method. Du et al. [36] developed a robot calibration method using a multi-sensor system based on their previous work [31]. All the calibration procedures were similar to [31], except, in this work, they used a Cerebellar Model Articulation Controller (MAC) algorithm to reduce the effect of noise on the accuracy of the pose estimation of the end-effector of the robot; moreover, an Unscented Kalman Filter (UKF) was used to estimate the errors between the actual and nominal kinematic parameters of the robot manipulator. The results showed an improvement in the accuracy of the new method.

In this paper, a novel robot calibration framework is proposed based on computer vision techniques using a fiducial marker system. Instead of a single marker as in [26], a map of ArUco markers is used for pose estimation. Using a map of markers solves the occlusion and camera FOV issues and enables the vision system to measure poses in different robot configurations. The use of a monocular camera instead of a stereo camera as in [24–26] is the main contribution of this paper. The stereo vision system uses two images per robot configuration to measure the target pose, increasing the computational complexity. Using monocular vision, simplifies the hardware of the calibration system and decreases the computational requirements. In addition to these, monocular vision obviated the synchronization and FOV issues present in stereo vision systems. One of the main features of the proposed calibration framework is that it is fully automatic and does not require any manual operation during calibration. It works with low-resolution images and high frame rates. Compared to the fusion-based techniques, such as CMM and LTS based systems [4,16,17,21,34], the proposed method is easier to use and does not require any complicated setups and auxiliary equipment. Overall, the proposed method is simple, cost-effective and yet satisfies the requirements of industrial robot calibration.

1.1. Overview of the proposed calibration method

A robotic manipulator is composed of a set of rigid links connected by means of joints [37]. One end of the robotic manipulator is fixed to the ground called the base and the other end of the robotic manipulator has an end-effector such as a gripper connected to it. A robot is programmed in such a way that its tool center point (TCP) follows a trajectory with prescribed poses (position and orientation) to perform a certain task. TCP is a programmable point that can be located anywhere on a robot [38], such as center of its end-effector or center of its last link. The trajectory of a TCP is dependent on the kinematic parameters of a robot that change over time. Therefore, these kinematic parameters need to be adjusted periodically using calibration. The calibration process consists of four steps: (i) kinematic modeling, (ii) pose measurement, (iii) kinematic identification, and (iv) kinematic compensation [39]. This paper focuses on the first three steps, as illustrated in Fig. 2.

1.2. Kinematic modeling using product-of-exponential formula

A kinematic model would be suitable for robot calibration if it satisfies the following three principles [40]:

1. Completeness: A complete model must have enough parameters to describe any possible deviation of the actual kinematic parameters from the nominal values.
2. Continuity: Mathematically, the kinematic model is a continuous and injective function of kinematic parameters. Any small variation in the geometry of a robot must correspond to a small variation in the kinematic parameters.

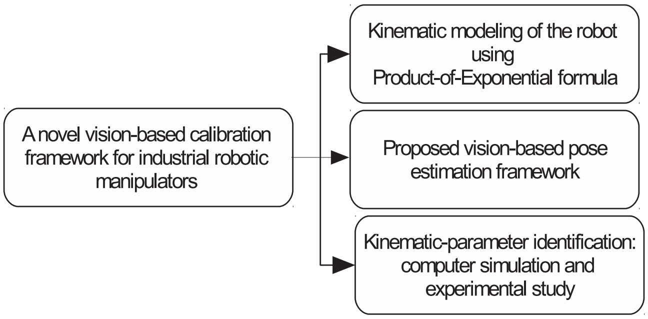

A novel vision-based calibration framework for industrial robotic manipulators
Kinematic modeling of the robot using Product-of-Exponential formula
Proposed vision-based pose estimation framework
Kinematic-parameter identification: computer simulation and experimental study

Fig. 2. Overview of the steps followed in the proposed calibration framework.

3. Minimality: “The kinematic model must include only a minimal number of parameters” [41]. The error model for the kinematic calibration should not have redundant parameters.

The standard D-H method has been used to model robot kinematics for many years; however, the error models based on the D-H methods are not continuous when two consecutive joint axes are close to being parallel [42]. Many researchers have suggested various models such as the Hayati et al. model [42,43], Veitschegger and Wu’s model [13], Stone and Sanderson’s S-model [44], and complete and parametrically continuous (CPC) model [39] to avoid the singularity problem of the standard D-H method. Nevertheless, these models are based on the D-H convention and they involve some redundant parameters. To eliminate the singularity problem, Park and Okamura used the product of exponential (POE) formula to calibrate a serial-robot [45]. They proposed a generic error model for calibration based on the POE formula developed by [46]. However, the expressions of their error model were not in the explicit form; moreover, they have not analyzed the identifiability of kinematic parameters in their error model [47]. In this paper, the error model based on the POE formula proposed in [47] is implemented. This error model is singularity free, and it also satisfies the three aforementioned principles.

The rest of the paper is organized as follows; in Section 2, the forward kinematic model using the POE method is introduced. In Section 3, the pose estimation framework is discussed. In Section 4, the calibration procedure is developed. In Section 5, the experimental procedure is introduced and the results are discussed. Finally, conclusions are made in Section 6.

2. Forward kinematic model using product of exponential formula

Brockett was the first to describe the forward kinematics of an open-chain serial robot using the product of the exponential of twists [46]. In this approach, two coordinate frames are needed, as shown in Fig. 3. The base coordinate frame, $\{B\}$, is stationary, and it is attached to any point of the robot provided that the point is fixed to the link 0. The tool coordinate frame, $\{T\}$, is attached to the end-effector of the robot or to its TCP. In order to formalize the forward kinematic of the robot manipulator based on the POE formula, the $T_{st}(0)$ is defined as the initial transformation from $\{T\}$ to $\{B\}$ when the robot is in reference configuration; meaning that all joint variables are equal to zero. Fig. 4 shows the UR5 robot manipulator in its reference or zero-position configuration. According to the POE-based forward kinematic model, the motion of each joint is generated by a twist, ξ, associated with the joint axis. Murray et al. [48] defined the twist coordinate of a revolute joint by:

$$\xi_i = \begin{bmatrix} v_i & \omega_i \end{bmatrix}^T = \begin{bmatrix} p_i \times \omega_i & \omega_i \end{bmatrix}^T \tag{1}$$

where ω_i is the unit directional vector of the ith revolute joint in the direction of the twist axis, and p_i is the coordinate of any point on the joint axis. All the twist coordinates are written in the base frame of the robot [47], provided that $\|\omega_i\| = 1$ be satisfied. If q_i is the joint variable of the ith joint, then a generic forward kinematic model for an n-degree-of-freedom (n-DOF) serial robot is given by:

$$T = \exp(\hat{\xi}_1 q_1)\exp(\hat{\xi}_2 q_2)\cdots\exp(\hat{\xi}_n q_n)T_{st}(0) \tag{2}$$

where $\hat{\xi}_i \in se(3)$ is expressed in the base frame as follows:

$$\hat{\xi}_i = \begin{bmatrix} \hat{\omega}_i & v_i \\ 0 & 0 \end{bmatrix} \tag{3}$$

where $v_i = [v_{1i}, v_{2i}, v_{3i}]^T \in \Re^3$ is the position vector of the ith prismatic joint, and $\hat{\omega}_i \in so(3)$ is the skew-symmetric matrix of $\omega_i = [\omega_{1i}, \omega_{2i}, \omega_{3i}]^T \in \Re^3$, and it is referred to as unit directional vector of the ith revolute joints. The $\hat{\omega}_i$ is given by:

$$\hat{\omega}_i = \begin{bmatrix} 0 & -\omega_{3i} & \omega_{2i} \\ \omega_{3i} & 0 & -\omega_{1i} \\ -\omega_{2i} & \omega_{1i} & 0 \end{bmatrix} \tag{4}$$

Park and Okamura [49] showed that, $T_{st}(0)$ can be written as follows:

$$T_{st}(0) = \exp(\hat{\xi}_{st}) \tag{5}$$

where $\hat{\xi}_{st}$ is called the twist of the initial transformation. Thus, Eq. (2) can be rewritten as follows:

$$T = \exp(\hat{\xi}_1 q_1)\exp(\hat{\xi}_2 q_2)\ldots\exp(\hat{\xi}_n q_n)\exp(\hat{\xi}_{st}) \tag{6}$$

2.1. The error model

The error model is calculated by linearizing the forward kinematic equation, Eq. (6), as follows [47,49]:

$$\delta TT^{-1} = \left(\frac{\partial T}{\partial \xi}\delta\xi + \frac{\partial T}{\partial \mathbf{q}}\delta\mathbf{q} + \frac{\partial T}{\partial \xi_{st}}\delta\xi_{st}\right)T^{-1} \tag{7}$$

where:

$$\xi = [\xi_1, \xi_2, \ldots, \xi_n]^T \in \Re^{6n} \tag{8}$$

$$\mathbf{q} = [q_1, q_2, \ldots, q_n]^T \in \Re^n \tag{9}$$

According to Eq. (7), the pose error of the TCP are due to the kinematic errors in ξ, ξ_{st}, and $\mathbf{q}$, which are denoted by $\delta\xi$, $\delta\xi_{st}$, and $\delta\mathbf{q}$, respectively. Let T_a and T_n be the actual and nominal pose of the TCP in Eq. (7), where T_a is obtained from the measurement data, and T_n is calculated from Eq. (6). Then, $\delta TT^{-1} \in se(3)$ would represent the deviation of T_a from T_n. If the deviation between the actual and nominal pose would be sufficiently small, then $T_a T_n^{-1}$ would be in the neighborhood of the identity in the group. Park and Okamura [49] showed that Eq. (7) can be rewritten as:

$$\delta TT^{-1} = \log(T_a T_n^{-1}) \tag{10}$$

After the computation of the Eq. (10), the essential next step is the identification of the kinematic parameters, which is the solution of the cost function:

$$\text{Min}\left\|\delta TT^{-1} - \left(\frac{\partial T}{\partial \xi}\delta\xi + \frac{\partial T}{\partial \mathbf{q}}\delta\mathbf{q} + \frac{\partial T}{\partial \xi_{st}}\delta\xi_{st}\right)T^{-1}\right\|^2 \tag{11}$$

He et al. [47] proposed Eq. (12) to compute δTT^{-1} in the Eq. (11).

$$\begin{aligned}
&[\delta TT^{-1}]^\vee \\
&= [(\delta\exp(\hat{\xi}_1 q_1))\exp(-\hat{\xi}_1 q_1)]^\vee + Ad(\exp(\hat{\xi}_1 q_1))[(\delta\exp(\hat{\xi}_2 q_2))\exp(-\hat{\xi}_2 q_2)]^\vee \\
&\quad + \cdots + Ad\left(\prod_{i=1}^{n-1}\exp(\hat{\xi}_i q_i)\right)[(\delta\exp(\hat{\xi}_n q_n))\exp(-\hat{\xi}_n q_n)]^\vee \\
&\quad + Ad\left(\prod_{i=1}^{n}\exp(\hat{\xi}_i q_i)\right)[(\delta\exp(\hat{\xi}_{st}))\exp(-\hat{\xi}_{st})]^\vee
\end{aligned} \tag{12}$$

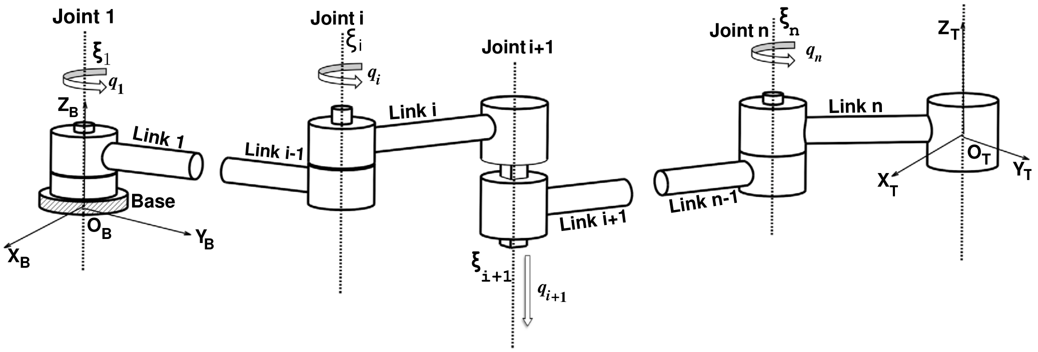


Fig. 3. Forward kinematics of an n-DOF serial robot.

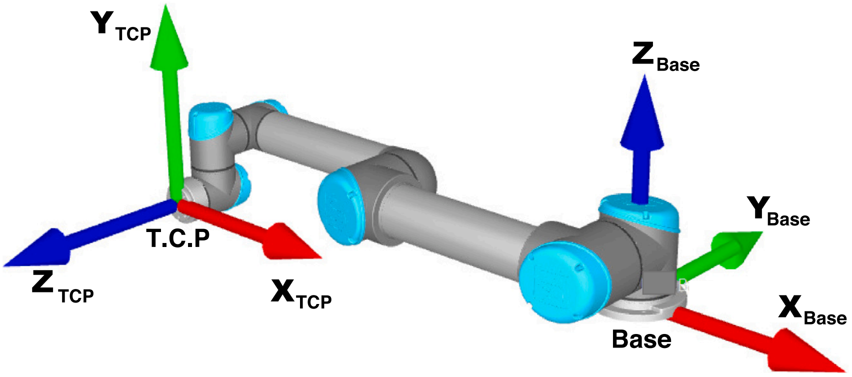


Fig. 4. UR5 with its TCP and base coordinate frames in zero-position configuration.

where:

$$[(\delta\exp(\hat{\xi}_i q_i))\exp(-\hat{\xi}_i q_i)]^{\vee} = \left(q_i \mathbf{I}_6 + \frac{4-\theta_i\sin(\theta_i)-4\cos(\theta_i)}{2\|\omega_i\|^2}\Omega_i + \frac{4\theta_i-5\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^3}\Omega_i^2 + \frac{2-\theta_i\sin(\theta_i)-2\cos(\theta_i)}{2\|\omega_i\|^4}\Omega_i^3 + \frac{2\theta_i-3\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^5}\Omega_i^4 \right)\delta\xi_i + \xi_i\delta q_i \qquad (13)$$

$$\Omega_i = \begin{bmatrix} \hat{\omega}_i & \mathbf{0} \\ \hat{v}_i & \hat{\omega}_i \end{bmatrix} \qquad (14)$$

$$\|\omega_i\| = (\omega_{1i}^2 + \omega_{2i}^2 + \omega_{3i}^2)^{1/2} \qquad (15)$$

$$\theta_i = \|\omega_i\| q_i \qquad (16)$$

and,

$$[(\delta\exp(\hat{\xi}_{st}))\exp(-\hat{\xi}_{st})]^{\vee} = \left(\mathbf{I}_6 + \frac{4-\theta_{st}\sin(\theta_{st})-4\cos(\theta_{st})}{2\theta_{st}^2}\Omega_{st} + \frac{4\theta_{st}-5\sin(\theta_{st})+\theta_{st}\cos(\theta_{st})}{2\theta_{st}^3}\Omega_{st}^2 + \frac{2-\theta_{st}\sin(\theta_{st})-2\cos(\theta_{st})}{2\theta_{st}^4}\Omega_{st}^3 + \frac{2\theta_{st}-3\sin(\theta_{st})+\theta_{st}\cos(\theta_{st})}{2\theta_{st}^5}\Omega_{st}^4 \right)\delta\xi_{st} \qquad (17)$$

$$\Omega_{st} = \begin{bmatrix} \hat{\omega}_{st} & \mathbf{0} \\ \hat{v}_{st} & \hat{\omega}_{st} \end{bmatrix} \qquad (18)$$

$$\|\omega_{st}\| = (\omega_{1st}^2 + \omega_{2st}^2 + \omega_{3st}^2)^{1/2} \qquad (19)$$

$$\theta_{st} = \|\omega_{st}\| \qquad (20)$$

The ($\vee$) operation maps $\hat{\xi}_i \in se(3)$ and $\hat{\omega}_i \in so(3)$ into $\xi_i \in \Re^6$ and $\omega_i \in \Re^3$ [45,49]. It is possible to denote the Eq. (12) as:

$$\mathbf{y} = \mathbf{J}\mathbf{x} \qquad (21)$$

where:

$$\mathbf{y} = \left[\delta T T^{-1}\right]^{\vee} \in \Re^6 \qquad (22)$$

$$\mathbf{J} = \left[\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_n, \mathbf{J}_{st}\right] \in \Re^{6\times(7n+6)} \qquad (23)$$

$$\mathbf{x} = \left[\delta\xi_1, \delta q_1, \delta\xi_2, \delta q_2 \ldots, \delta\xi_n, \delta q_n, \delta\xi_{st}\right]^T \in \Re^{7n+6} \qquad (24)$$

Where **J** is the identification Jacobian matrix, **y** is the pose error vector of the TCP expressed in the base coordinate frame of the robot, and **x** is the error vector of the kinematic parameters. The elements of the identification Jacobian matrix are calculated as follows:

$$\mathbf{J}_i = \begin{cases} [A_1, \xi_1], & i = 1 \\ Ad\left(\prod_{k=1}^{i-1}\exp(\hat{\xi}_k q_k)\right)[A_i, \xi_i], & 1 < i < n+1 \\ Ad\left(\prod_{k=1}^{i-1}\exp(\hat{\xi}_k q_k)\right)A_{st}, & i = n+1 \end{cases} \qquad (25)$$

2.2. Kinematic parameters identification

In order to solve Eq. (21), it is required to make the Jacobian matrix, **J**, identifiable, which means that its columns are not linear combinations of each other; otherwise, the redundant-parameters issue would occur [50,51]. This requirement would be satisfied by making the **J** matrix full column rank. He et al. [47] proved that $\delta\mathbf{q}$ and

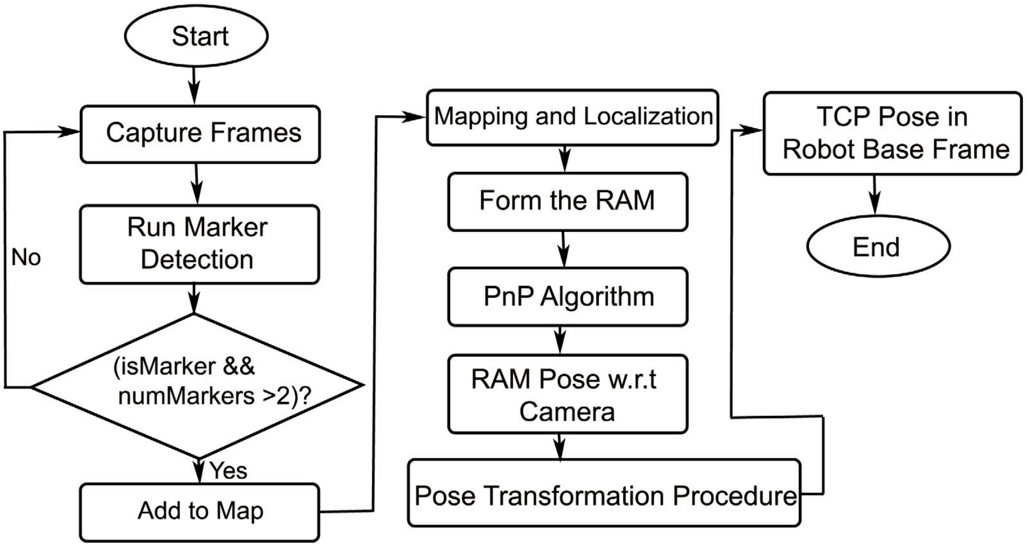


Fig. 5. The flowchart of the proposed calibration framework for the pose estimation of the TCP.

$\delta\xi_{st}$ could be removed from the error model, Eq. (7); otherwise, the redundant-parameters issue would occur in the computations of the error models. By this assumption, the error model in Eq. (7), and Eqs. (24)–(25) are simplified as follows [26,47,52]:

$$\delta T T^{-1} = \left(\frac{\partial T}{\partial \xi} \delta \xi \right) T^{-1} \tag{26}$$

$$\mathbf{J}_i = \begin{cases} A_1, & i = 1 \\ Ad\left(\prod_{k=1}^{i-1} \exp(\hat{\xi}_k q_k) \right) A_i, & 1 < i < n+1 \\ Ad\left(\prod_{k=1}^{i-1} \exp(\hat{\xi}_k q_k) \right) A_{st}, & i = n+1 \end{cases} \tag{27}$$

$$\mathbf{x} = \left[\delta\xi_1, \delta\xi_2, \ldots, \delta\xi_n, \delta\xi_{st} \right]^T \in \Re^{6n+6} \tag{28}$$

To reduce the effect of measurement noise and also to make the identification Jacobian matrix, $\mathbf{J}$, full column-rank, m measurements are required with the following condition [26]:

$$m \geq n + 1 \tag{29}$$

where m is the number of pose measurements, and n is the degree-of-freedom of the robot. After taking m measurements, the matrix representation of Eq. (21) becomes:

$$\begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^m \end{bmatrix} = \begin{bmatrix} \mathbf{J}^1 \\ \vdots \\ \mathbf{J}^m \end{bmatrix} \mathbf{x} \tag{30}$$

The compact form of Eq. (30) is given by:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{J}} \mathbf{x} \tag{31}$$

$\mathbf{x}$ can be solved by the iterative least-square algorithm [40,49,52], Eq. (32).

$$\mathbf{x} = (\tilde{\mathbf{J}}^T \tilde{\mathbf{J}})^{-1} \tilde{\mathbf{J}}^T \tilde{\mathbf{y}} \tag{32}$$

3. Pose estimation framework

The pose estimation framework is based on ArUco markers [53] that are used as fiducial markers to estimate the pose of the TCP. In this framework, a set of ArUco markers are used in order to increase the number of visible markers by the camera. After the detection of all visible markers, mapping and localization of these markers are performed. Then, the pose of the TCP is estimated. This process is illustrated in Fig. 5.

3.1. ArUco markers

ArUco markers, Fig. 6, are used as the fiducial markers due to their efficient and robust detection algorithm. A single fiducial marker provides four 3D to 2D correspondence points that can be used to localize the pose of the calibrated camera with respect to the marker [54]. However, using a single fiducial marker causes problems for pose estimation. The first problem is associated with pose ambiguity. In this case, two solutions are calculated for the pose of the camera in the coordinate frame of the marker, and it is impossible to specify which solution is the correct answer [55]. The second problem is related to occlusion. Some parts of the marker might be occluded causing the marker not to be detected. A set of ArUco markers, Fig. 6, referred to as a map of markers, is used to avoid the occlusion problems of the single markers. Inspired by [56], a truncated-rhombicuboctahedron shape, a convex polyhedral, was selected as the structure of the set of markers. The truncated-rhombicuboctahedron shape has thirteen squares and four isosceles triangles; its facets are shown in Fig. 6. This structure is manufactured using a 3D printer and mounted on the TCP of a robotic manipulator. ArUco markers are arranged in such a way that at every pose of the TCP, the minimum number of detected markers is four, Fig. 6(b).

3.2. Geometric camera modeling and calibration

Before the detection step, the camera is calibrated using the ChArUco pattern [57], Fig. 7. During camera calibration, intrinsic parameters (focal length and optical centers), extrinsic parameters, and distortion coefficients are determined at the three different resolutions.

3.3. Marker detection, mapping and localization

The following procedures are performed to detect each ArUco marker on the rhombicuboctahedron structure, Fig. 8(a), as follows: (i) image segmentation is performed using adaptive thresholding, (ii) contours are extracted, and the contours that are not convex or not an approximate square are filtered out, (iii) marker ID is extracted, and (iv) sub-pixel corner refinement is performed to detect all

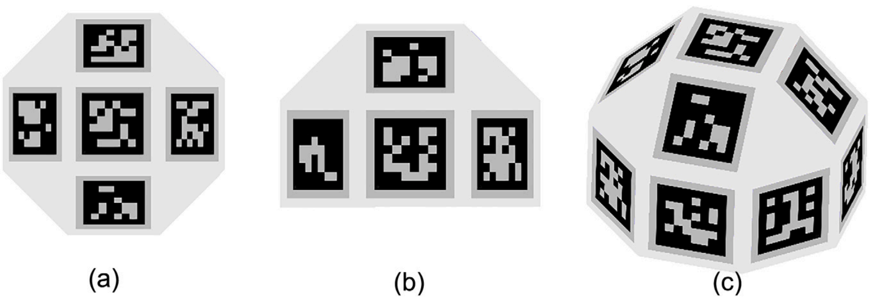


Fig. 6. ArUco markers. (a) Top-view, (b) Side-view, (c) Iso-metric view.

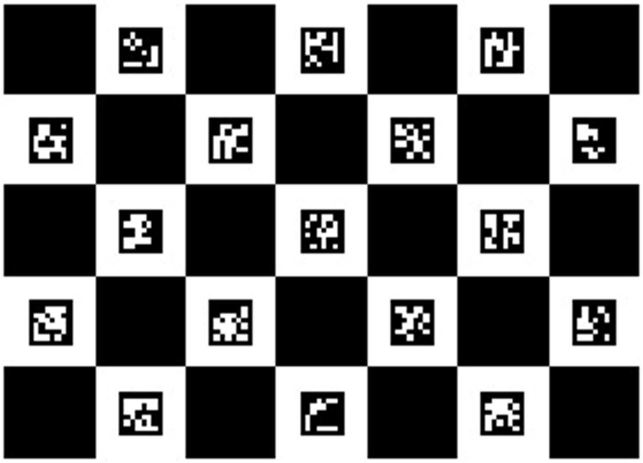


Fig. 7. ChArUco calibration board [57].

four corners of a marker accurately [58]. In the following, mapping and localization algorithm are performed using the method proposed in [54]. In this method, first, an approximate pose graph is formed where nodes represent the position of markers, and edges represent the relative distance between the markers. The graph is then corrected by propagating the error to each of its nodes. Finally, the corrected graph is optimized using the Levenberg–Marquardt algorithm so that 3D to 2D pose projection of the four corners of ArUco markers is estimated with minimal error with respect to the camera reference frame, Fig. 5. From this point on, the localized and mapped markers on the rhombicuboctahedron is called the RAM, which stands for **R**hombicuboctahedron **A**rUco **M**ap, Fig. 8(b).

3.4. RAM pose estimation

Perspective-n-Point (PnP) algorithm is used to estimate the 6-DOF pose of the camera with respect to the RAM reference frame (the reference frame of the RAM is defined as the coordinate axes of Marker#0) [59]. In the PnP algorithm, the 3D to 2D correspondence between the four corners of an ArUco marker in the reference frames of the RAM and camera is used to estimate the pose of the camera with respect to the RAM using the following equation:

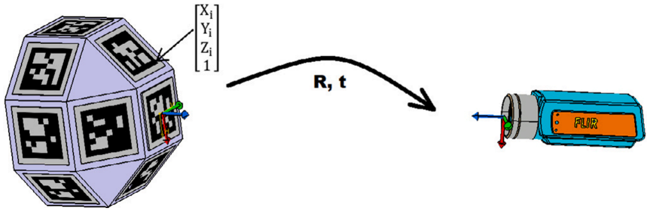


Fig. 9. Illustration of the PnP algorithm.

$$s\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{33}$$

$$[\mathbf{R} \quad | \quad \mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{34}$$

where **R** and **t** are the rotation matrix and translation vector between the RAM and the camera coordinate frames, respectively. X_i, Y_i, and Z_i are the coordinates of the corners of ArUco marker in the RAM reference frame, and u_i, v_i are their corresponding 2-D coordinates in the camera reference frame. c_x, c_y, f_x, and f_y are the camera principle points and camera focal lengths, respectively, and **s** is the scaling factor. Using Eqs. (33) and (34), the position and orientation between camera and RAM reference frame are calculated as:

$$s\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}[\mathbf{R} \quad | \quad \mathbf{t}]\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{35}$$

The schematic representation of the PnP algorithm is illustrated in Fig. 9.

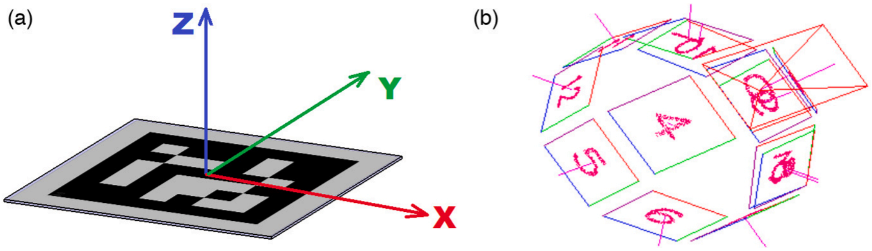


Fig. 8. Detection of ArUco markers. (a) Reference frame of marker #0, and (b) Mapped and localized Rhombicuboctahedron (RAM).

3.5. TCP pose estimation in the base coordinate frame of the robot

The pose of the camera is known on the reference frame of the RAM up to this point. In the following, using that information, homogeneous transformations are carried out to estimate the pose of the TCP in the base coordinate frame of the robot, as illustrated in Fig. 10.

In Fig. 10, $\hat{H}_{XY}^{(i,j)}$ represents the homogeneous transformation matrix that transforms the coordinate frame with orientation j and origin Y to a coordinate frame with orientation i and origin X. The homogeneous transformation from the TCP to the base coordinate frame is given as:

$$\hat{H}_{BT}^{(base,TCP)} = \hat{H}_{BC}^{(base,Cam)} \hat{H}_{CR}^{(Cam,RAM)} \hat{H}_{RT}^{(RAM,TCP)} \tag{36}$$

where, $\hat{H}_{RT}^{(RAM,TCP)}$ is the homogeneous transformation matrix from the TCP to the RAM. $\hat{H}_{RT}^{(RAM,TCP)}$ is a constant matrix and it is determined when the rhombicuboctahedron structure is fixed to the robotic manipulator. $\hat{H}_{CR}^{(Cam,RAM)}$ is the homogeneous transformation matrix from the RAM to the camera. It is calculated using the PnP algorithm, hence it is a known parameter. $\hat{H}_{BC}^{(base,Cam)}$ is the hand–eye matrix or the homogeneous transformation matrix from the camera coordinate frame to the base coordinate frame of the robot. It is calculated using the hand–eye calibration method of the industrial machine vision software, HALCON [60]. Finally, $\hat{H}_{BT}^{(base,TCP)}$ is the homogeneous transformation matrix from the TCP to the base coordinate frame of the robot, and it is used to estimate the pose of the TCP in the base coordinate frame.

4. Results

The performance of the proposed calibration framework was evaluated systematically through simulation-based and real-world experiments using two different metrics.

4.1. Metrics

Accuracy and repeatability metrics were used to evaluate the performance of the proposed calibration framework. Accuracy was used for both position and orientation measurements based on [61], and they are defined as follows:

$$\Delta P = \frac{1}{m}\sum_{i=0}^{m} \| P_{ai} - P_{ci} \| \tag{37}$$

$$\Delta R = \frac{1}{m}\sum_{i=0}^{m} \| \log(R_{ai}^{-1} R_{ci})^{\vee} \| \tag{38}$$

Where m is the number of measurements, ΔP and ΔR are the average position and orientation accuracy, respectively. The pairs (R_{ai}, P_{ai}) and (R_{ci}, P_{ci}) denote the actual and calibrated kinematic parameters at a given ith pose, respectively.

Repeatability was also used for both position and orientation measurements. The position repeatability of the robot at a given ith pose is calculated as follows [3]:

$$RP_i = \frac{1}{n}\sum_{j=1}^{n} P_j + 3\sqrt{\frac{\sum_{j=1}^{n}(P_j - \bar{P})^2}{n-1}} \tag{39}$$

P_j and $\bar{P}$ are calculated as follow:

$$P_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \tag{40}$$

$$\bar{P} = \frac{1}{n}\sum_{j=1}^{n} P_j \tag{41}$$

Where x_j, y_j, and z_j are the measured position of the TCP at jth cycle on the X, Y, and Z axes, respectively, and $\bar{x}$, $\bar{y}$, and $\bar{z}$ are their corresponding mean values. Similarly, the orientation repeatability of the robot at a given ith pose, RP_i^a, RP_i^b, and RP_i^c are calculated about the X, Y, and Z axes, respectively:

$$RP_i^a = \pm 3\sqrt{\frac{\sum_{j=1}^{n}(a_j - \bar{a})^2}{n-1}} \tag{42}$$

$$RP_i^b = \pm 3\sqrt{\frac{\sum_{j=1}^{n}(b_j - \bar{b})^2}{n-1}} \tag{43}$$

$$RP_i^c = \pm 3\sqrt{\frac{\sum_{j=1}^{n}(c_j - \bar{c})^2}{n-1}} \tag{44}$$

Where a_j, b_j, and c_j are the measured orientation of the TCP at a given pose ith of jth cycle, about the X, Y, and Z axes, respectively, and $\bar{a}$, $\bar{b}$, and $\bar{c}$ are their correspondent mean values. The position and orientation measurements of the TCP are taken using the proposed vision system described in Section 3.

4.2. Calibration experiments

Calibration experiments were performed both in simulation and real-world environments to substantiate the efficiency and robustness of the proposed calibration framework. In both experiments, the twist coordinates of the revolute joints were calculated in the zero-position configuration of the robot, Fig. 4. The POE-based robot calibration is an iterative procedure [40,49,61], Fig. 11. In this approach, the first operation is the POE-based kinematic modeling of the robot. For this purpose, the twist coordinates of joints are calculated according to Eq. (1) for both the nominal and actual kinematic models of the robot, ξ_n and ξ_a. The ξ_n and ξ_a are also referred to as the nominal and actual kinematic parameters. The nominal kinematic parameters are shown in Table 1. In the second step, (i) nominal and actual poses of the TCP, T_n and T_a, (ii) pose errors of the TCP, and (iii) the Jacobian matrices of each revolute joint in m different configurations of the robot are calculated by substituting the calculated ξ_ns' and ξ_as' into Eqs. (6), (26), and (27). Finally, by forming Eq. (31) and solving it using the iterative least-square algorithm, Eq. (32), the kinematics parameter identification step would be accomplished.

According to Fig. 11, the identified kinematic parameters are needed to be updated in the iterative calibration method. Let $\mathbf{P}_{i-1}$ be the kinematic parameters being computed at the $(i-1)$th iteration; then, the updated kinematic parameters at the ith iteration would be estimated using:

$$\mathbf{P}_i = \mathbf{P}_{i-1} + \mathbf{x} \tag{45}$$

The updated kinematic parameters would be used for the calculation of the calibrated poses, T_c, at the subsequent iterations. At the end of each iteration, the pose error vector, Eq. (31), is computed between the actual and calibrated poses using Eqs. (37)–(38). The pose error vector, Eq. (31), is used as a decision criterion for stopping or continuing of the calibration iterations. Iterations continue till all elements converge to zero, and the elements of $\mathbf{P}$ become stable with the increasing number of iterations. The use of POE-based kinematic modeling in the proposed calibration framework made it singularity free, which was unavoidable in the D-H methods.

4.2.1. Simulation-based experiments

Simulation-based experiments were performed using a PC to evaluate the effects of noise and to evaluate the robustness of the calibration procedure. As shown in Fig. 11, the nominal and actual poses of the TCP are the inputs of the calibration procedure. The actual pose of the TCP is obtained by the substitution of the actual twist coordinates of the joints, Eq. (1), into Eq. (6). Let the nominal twist coordinate, ξ_n, is represented by Eq. (1). Then, the actual twist coordinates, ξ_a, can be simulated by adding preset kinematic errors, $\delta\omega$ and δv, to the nominal twist coordinates of joints as follows [61]:

$$\begin{bmatrix} \omega_i + \delta\omega_i \\ v_i + \delta v_i \end{bmatrix} \tag{46}$$

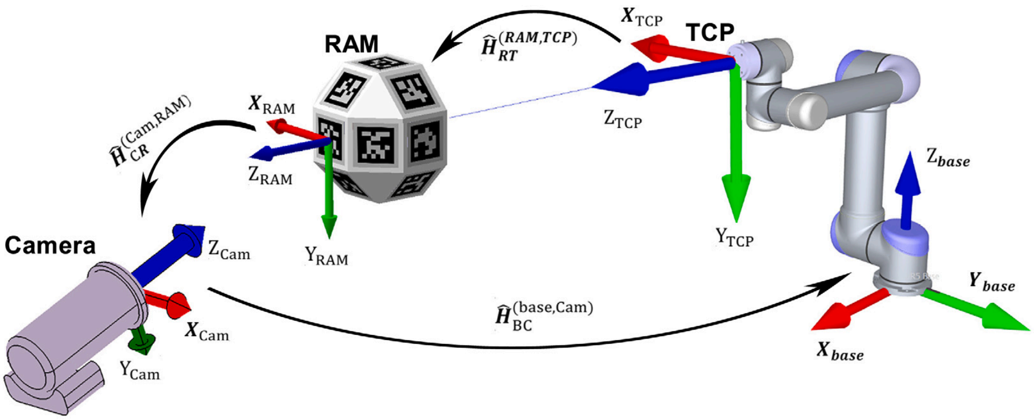


Fig. 10. Homogeneous transformations from TCP to the Robot base coordinate frame.

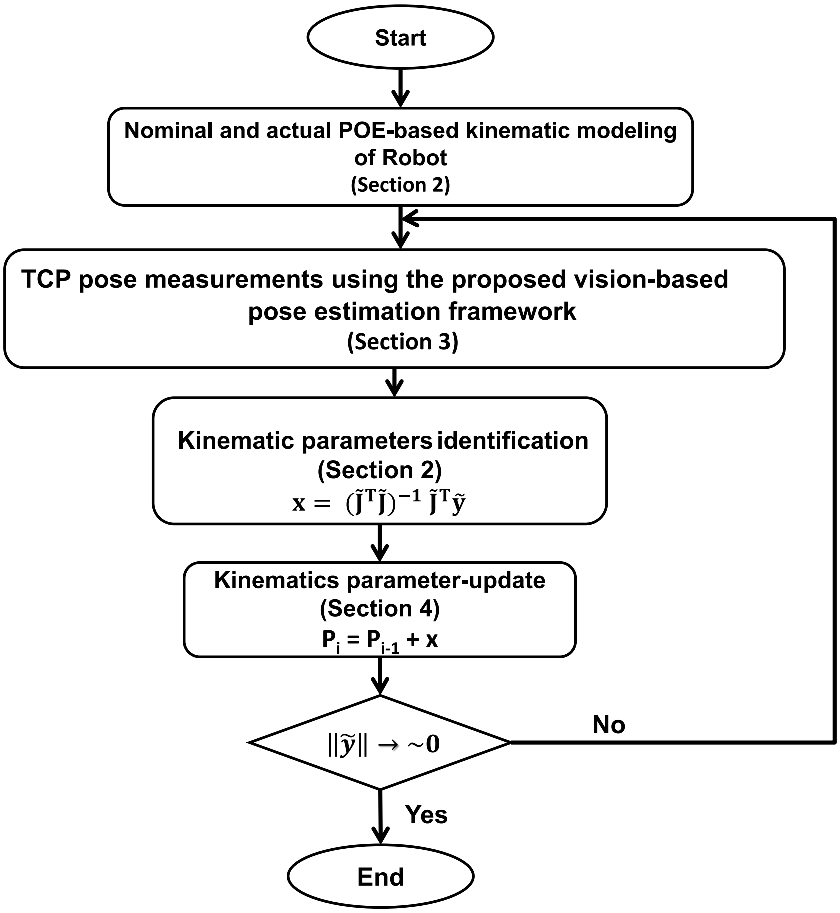

Start
Nominal and actual POE-based kinematic modeling of Robot
(Section 2)
TCP pose measurements using the proposed vision-based pose estimation framework
(Section 3)
Kinematic parameters identification
(Section 2)
$\mathbf{x} = (\tilde{\mathbf{J}}^T\tilde{\mathbf{J}})^{-1}\tilde{\mathbf{J}}^T\tilde{\mathbf{y}}$
Kinematics parameter-update
(Section 4)
$P_i = P_{i-1} + x$
$\|\tilde{y}\| \rightarrow \sim 0$
No
Yes
End

Fig. 11. Flowchart of the calibration process.

For the calculation of the preset kinematic errors, two necessary conditions are specified as follows [61]:

(1) $\|\omega_i + \delta\omega_i\| = 1$
(2) $(\omega_i + \delta\omega_i)^T (v_i + \delta v_i) = 0$

Simulation-based experiments were performed using 50 simulated poses of the TCP for the calibration phase, and other 50 poses for the validation of the calibration results. Fig. 12 shows the results of the simulation-based experiments. It can be observed that the position and orientation errors became almost zero after the second iteration. The updated kinematic parameters in simulation and real-world experiments are typically referred to as the calibrated kinematic parameters, Table 1.

4.2.2. *Real-world experiments*

In real-world experiments, a 6-DOF UR5 robot was used to evaluate the performance of the proposed calibration framework. In these experiments, the actual pose of the TCP, which is required for the calibration procedure, is measured using the proposed vision-based system described in Section 3. The experimental setup consisted of a map of the ArUco markers in a rhombicuboctahedron shape and a Logitech C922-pro web camera as shown in Fig. 13. The measured

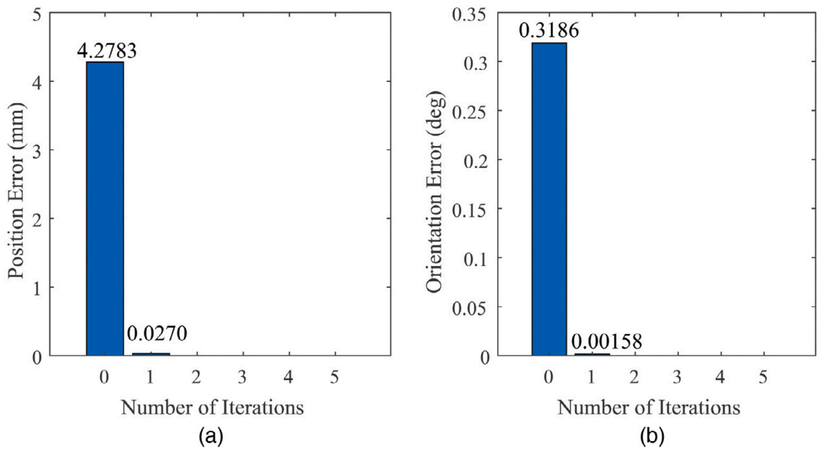


Fig. 12. Performance of the simulation-based experiments. (a) Position error, and (b) Orientation error.

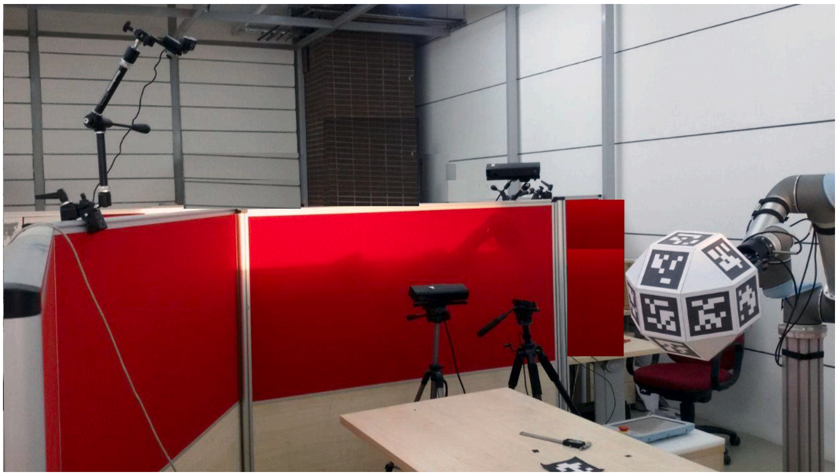


Fig. 13. The setup for real-world experiments.

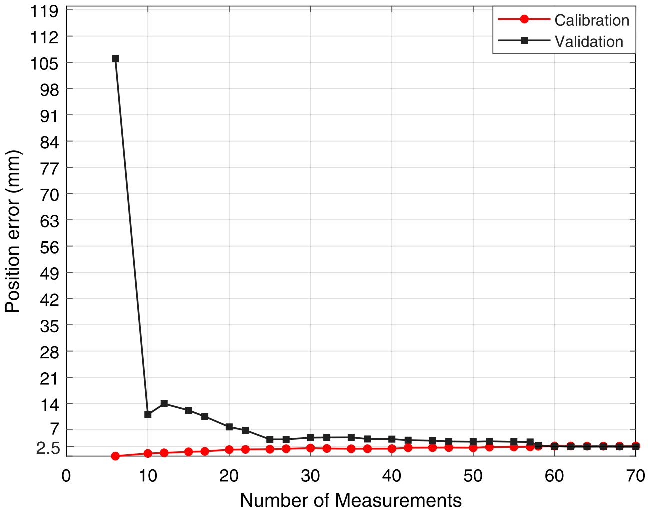


Fig. 14. Calibration results for real-world experiments: Absolute Position error.

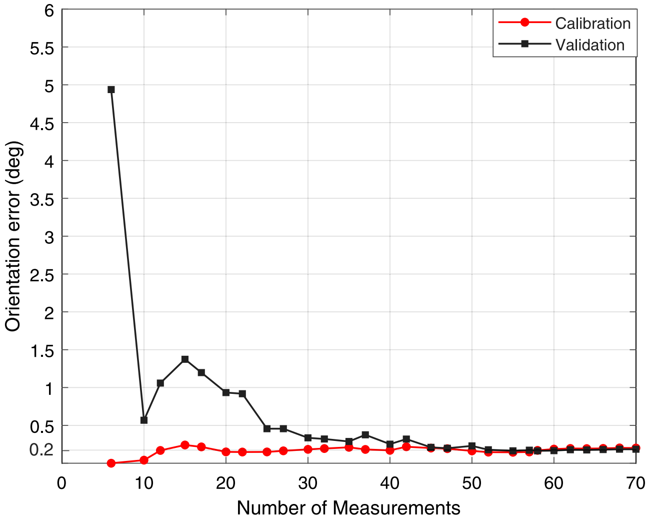


Fig. 15. Calibration results for real-world experiments: Absolute orientation error.

poses of the TCP are grouped into two. The first group had 70 poses used for calibration. The calibration group ($\mathbb{S}$) was subdivided into 27 different sets, each set having a different number of poses defined as:

$$\mathbb{S} = \{6, 10, 12, 15, 17, 20, 22, 25, 27, 30, 32, 35, 37, 40, \\ 42, 45, 47, 50, 52, 55, 57, 60, 62, 64, 66, 68, 70\}$$

Table 1
The nominal, actual and calibrated values of each joint of the UR5 robot. The calibrated values are calculated in simulation-based experiments.

Kinematic parameters	Nominal values	Actual values	Calibrated values
ξ_1	$\begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0.002709\\0.00192\\1.00024\\7.004\times10^{-5}\\2.500\times10^{-4}\\1.859\times10^{-4}\end{bmatrix}$	$\begin{bmatrix}0.002709\\0.001924\\1.000239\\0.00007\\0.000245\\0.000187\end{bmatrix}$
ξ_2	$\begin{bmatrix}0\\-1\\0\\0.089159\\0\\0\end{bmatrix}$	$\begin{bmatrix}0.00192\\-0.9989\\-4.005\times10^{-5}\\0.089142\\4.7952\times10^{-5}\\-6.1627\times10^{-6}\end{bmatrix}$	$\begin{bmatrix}0.001933\\-0.99895\\-0.00004\\0.089141\\0.000046\\-0.0000047\end{bmatrix}$
ξ_3	$\begin{bmatrix}0\\-1\\0\\0.089159\\0\\0.425\end{bmatrix}$	$\begin{bmatrix}4.6469\times10^{-4}\\-0.9991\\-7.4561\times10^{-4}\\0.08956\\-4.5167\times10^{-4}\\0.42520\end{bmatrix}$	$\begin{bmatrix}0.000469\\-0.99913\\-0.000739\\0.089556\\-0.000448\\0.425202\end{bmatrix}$
ξ_4	$\begin{bmatrix}0\\-1\\0\\0.089159\\0\\0.81725\end{bmatrix}$	$\begin{bmatrix}-0.003328\\-1.00118\\-5.5612\times10^{-4}\\0.08924\\-3.3404\times10^{-4}\\0.817344\end{bmatrix}$	$\begin{bmatrix}-0.003321\\-1.001185\\-0.000562\\0.089238\\-0.000340\\0.817347\end{bmatrix}$
ξ_5	$\begin{bmatrix}0\\0\\-1\\0.10915\\-0.81725\\0\end{bmatrix}$	$\begin{bmatrix}-0.00242\\1.3238\times10^{-4}\\-0.99869\\0.109366\\-0.81738\\5.1411\times10^{-5}\end{bmatrix}$	$\begin{bmatrix}-0.002424\\0.000133\\-0.99868\\0.109365\\-0.81738\\0.0000492\end{bmatrix}$
ξ_6	$\begin{bmatrix}0\\-1\\0\\-0.00445\\0\\0.81725\end{bmatrix}$	$\begin{bmatrix}-0.00242\\-0.99869\\1.3238\times10^{-4}\\-0.00446\\1.8222\times10^{-4}\\0.817368\end{bmatrix}$	$\begin{bmatrix}-0.002418\\-0.998691\\0.0001326\\-0.004461\\0.0001811\\0.817372\end{bmatrix}$

The calibration procedure was performed iteratively in each set of the calibration group according to Fig. 11. The second group was referred to as the validation group that consisted of the 30 poses to evaluate calibration result for each calibration set. Similar to simulation-based experiments, the position and orientation accuracy was calculated using Eqs. (37) and (38), and the results are depicted in Figs. 14 and 15. The absolute position and orientation errors for the calibration and validation are converged to 2.5 mm and 0.2°, respectively, Figs. 14 and 15.

Figs. 16(a), (b), and (c) show position errors of the TCP measured along the X, Y, and Z axes in the base coordinate frame of the robot, respectively. According to these plots, as the number of poses increases for the calibration, the position errors converge to 1.7 mm, 1.3 mm, and 1.4 mm along the X, Y, and Z axes, respectively. Similarly, Figs. 17(a), (b), and (c) show the orientation errors measured about the X, Y, and Z axes in the base coordinate frame of the robot. The orientation errors about the X, Y, and Z axes are converged to 0.063°, 0.074°, and 0.17°, respectively.

4.3. *Repeatability experiment*

Repeatability experiments were performed in a real-world setting. Five different pre-specified robot configurations (A, B, C, D, and E) were used for these experiments. These points were specified such that when the TCP reached one of these points, at least seven ArUco markers could be detectable by the proposed vision-based pose estimation system, Section 3. The pose repeatability of the proposed calibration framework was calculated in thirty cycles according to the TCP's movement pattern depicted in Fig. 18.

The position and orientation repeatability of the robot at points A to E are calculated using Eq. (39) and Eqs. (42)–(44), respectively. Table 2 shows the results of the repeatability experiments of the robot.

According to Table 2, the orientation and position repeatability of the robot measured by the proposed calibration framework was approximately zero and 0.4–0.8 mm, respectively.

5. Discussion

In this section, factors that affect the performance of the proposed calibration framework are addressed in depth. First, some aspects of the POE kinematic model, and then the parameters that affected the pose accuracy of the proposed calibration framework are discussed.

5.1. *Robot calibration based on the POE kinematic model*

The application of the POE kinematic model saved the proposed calibration framework from the singularity issues that might have been experienced if the D-H method was used for modeling. In the simulation-based experiments, the calibration errors fully recovered, and the absolute position error decreased from 4.2783 mm to 0.0270 mm, and the absolute orientation error decreased from 0.3186° to 0.00158° in the second iteration during calibration, showing the effectiveness of the kinematic calibration method.

Real-world experiments were conducted to investigate the effectiveness of the calibration framework with real hardware. In these experiments, the absolute position and orientation errors converged to 2.5 mm and 0.2°, respectively as shown in Figs. 14 and 15. The errors became stable with the increase in the pose numbers as expected. Thus, these experiments validated the applicability of the POE-based kinematic modeling in the proposed calibration framework for real-life scenarios.

5.2. *The effect of hardware on pose estimation performance*

A computer and a camera are the two main components used in the proposed calibration framework. This section briefly discusses the effects of computer configuration and the camera specifications on the calibration performance.

5.2.1. *Computer*

Image segmentation and marker code extraction are the two computationally intensive processes that take the most processing time of detecting the ArUco markers of the map. As the image size increases, this issue gets worse [62]. The experiments were performed using a computer with an Intel® Core™ i7-3470 with 8 GB RAM. The processing time of the proposed calibration framework for the pose estimation of the TCP was measured to be between 19 ms to 25 ms during the calibration procedure, which depended on the number of detected ArUco markers at the resolution of 1280 × 720. These results showed that a standard PC is appropriate for the calibration framework. However, for high-resolution images and higher frame rates, a powerful computer would be necessary. Higher resolutions were needed when the distance between the camera and the robot was increased. Similarly, higher frame rates were required for the online calibrations.

5.2.2. *Camera*

Type of camera, camera calibration and resolution are the three determining factors in calibration performance. The camera used in this study was equipped with a CMOS sensor. Although CMOS sensors are high-speed imagers suited for detecting fast-moving objects, they are susceptible to noise and have large distortion coefficients compared to CCD sensors. The noise and lens distortion caused erroneous detection of markers affecting the performance of pose estimation adversely. The lens distortion of the camera was computed using camera calibration. The camera was calibrated using a ChArUco board precisely

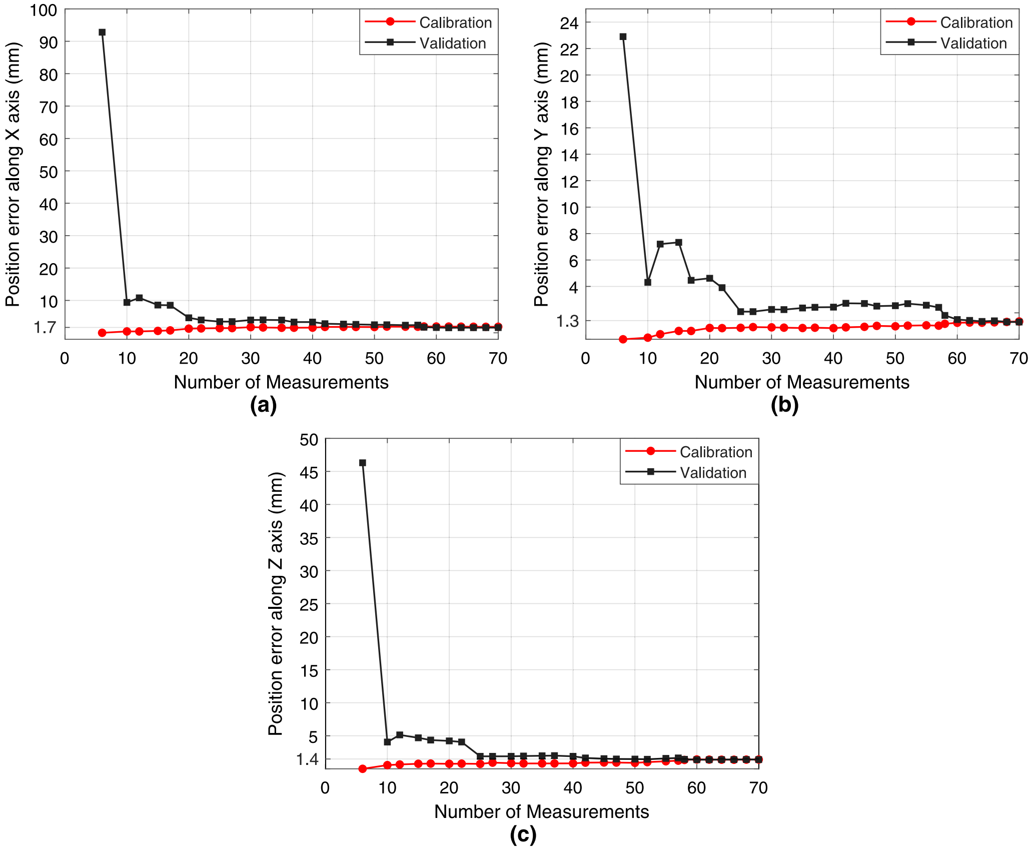


Fig. 16. Calibration results for real-world experiments. (a) Position error along the X-axis, (b) Position error along the Y-axis, and (c) Position error along the Z-axis.

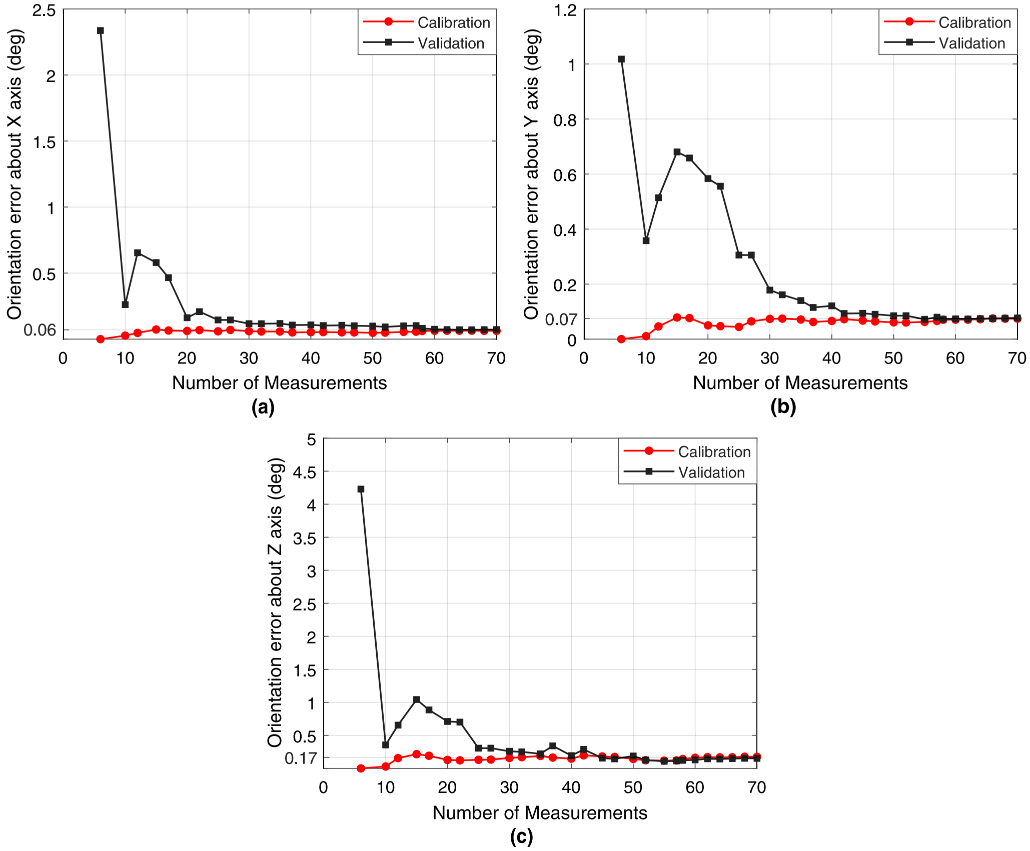


Fig. 17. Calibration results for real-world experiments. (a) Orientation error about X-axis, (b) Orientation error about Y-axis, and (c) Orientation error about Z-axis.

Table 2
Results of the position and orientation repeatability experiments of the robot.

Configuration	$PR(mm)$	$OR_a(°)$	$OR_b(°)$	$OR_c(°)$
A	0.8253	$\pm 2.6125 \times 10^{-13}$	$\pm 0.9500 \times 10^{-13}$	$\pm 0.9500 \times 10^{-13}$
B	0.4178	$\pm 1.7021 \times 10^{-13}$	$\pm 4.9083 \times 10^{-13}$	$\pm 0.7521 \times 10^{-13}$
C	0.8086	$\pm 2.2167 \times 10^{-13}$	$\pm 2.6917 \times 10^{-13}$	$\pm 2.2958 \times 10^{-13}$
D	0.6337	$\pm 1.3458 \times 10^{-13}$	$\pm 5.2250 \times 10^{-13}$	$\pm 0.1088 \times 10^{-13}$
E	0.5413	$\pm 1.4250 \times 10^{-13}$	$\pm 0.55417 \times 10^{-13}$	$\pm 1.4250 \times 10^{-13}$

Table 3
Distortion coefficients and reprojection error of camera at different resolutions.

Resolution	k_1	k_2	k_3	p_1	p_2	Reprojection error
640 × 480	0.0507	−0.1561	−0.0028	−0.0025	0.0329	0.1763
1280 × 720	0.0497	−0.1873	−0.0026	0.0032	0.1023	0.1958
1920 × 1080	0.1069	−0.3740	0.0056	−0.0032	1.0518	0.2160

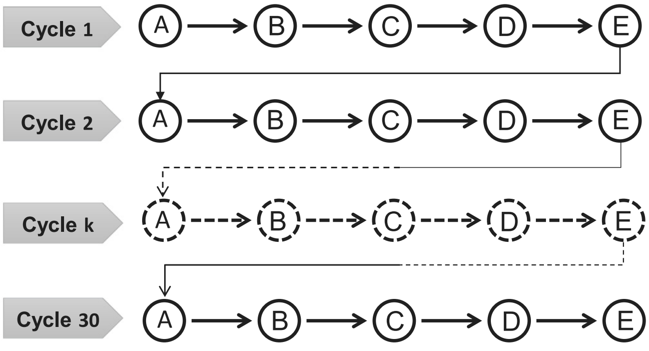


Fig. 18. Robot movement pattern implemented for the repeatability experiment.

for its intrinsic and extrinsic parameters. Three different resolutions: (i) 640 × 480, (ii) 1280 × 720, and (iii) 1920 × 1080 were used during camera calibration operations. Table 3 shows the distortion coefficients and the reprojection errors measured during camera calibration at the three different resolutions. It was observed that with an increase in the resolution of the camera, the reprojection error becomes larger, which negatively affected the accuracy of the camera matrix. The smallest reprojection error was calculated for the resolution 640 × 480, but the detection of corners of ArUco markers became difficult at that resolution. At the highest resolution, 1920 × 1080, the detection time and distortion coefficients were grown considerably, slowing down the system and affecting the accuracy of pose results. Based on the calibration results, 1280 × 720 was observed to be the best resolution for the proposed calibration framework.

5.2.3. ArUco

The optimized version of ArUco markers, referred to as ArUco.3, were used. In this section, some aspects of the ArUco markers that affected the accuracy of the proposed calibration framework are discussed.

I. Evaluation of the ArUco markers. Three essential metrics were defined by [63] to evaluate the reliability and robustness of the fiducial markers as follows:

- *False-positive rate*: It is defined as detecting a non-existing marker's ID while there is no marker with detected ID.
- *Inter-marker confusion rate*: This metric describes the possibility of the fiducial marker system to detect the IDs of the markers wrongly.
- *False-negative rate*: In this case, despite the presence of a fiducial marker in the scene, the fiducial marker system cannot identify and report the existence of that marker.

None of the abovementioned cases indicated by the metrics has been observed during experiments, guaranteeing the reliability of ArUco markers used in the framework.

II. Vertex jitter. Generally, the vertex-jitter [62] is a significant problem in fiducial marker systems. In such situations, the ArUco marker's corners flicker during marker detection step. These flickers lead to erroneous results in pose estimation. Despite the efforts to eliminate vertex-jitter, this undesirable effect was sometimes observed during the experiments.

III. Speed and memory. During the experiments, the detection of ArUco markers was fast enough, and no memory problem was observed; thus, there were no limitations in using oversized markers or more markers in the RAM structure. This property was due to using mixed-integer linear programming algorithms in the design of the ArUco.3 marker system [53].

IV. Illumination. Illumination was the main source of noise in real-world experiments. During the experiments, the illumination level of the environment was tried to be controlled as much as possible. However, non-uniform illumination of the environment due to the fluorescent lamps and daylight further increased the vertex-jitter and blinking-marker effects, decreasing the marker detection performance. Fig. 19(a) and (b) shows a case in which marker #9 is in blinking mode, which causes errors in estimating the pose of TCP. It was not possible to change the environmental conditions due to real-world limitations; instead, more measurements were made to eliminate these effects.

5.3. Robot-camera calibration matrix

A considerable portion of pose estimation errors was due to the erroneous hand–eye calibration matrix. The accuracy of the hand–eye matrix strongly depends on the quality of the images captured by the vision system [64]. Since a regular web camera with a CMOS sensor was used in this paper, the accuracy of the hand–eye matrix was low, and this deficiency propagated to the pose estimation of the TCP using Eq. (36).

5.4. Temperature and load

When a robot works under high temperature or load, some deflections will occur in the links of the robot that would lead to positional inaccuracies. According to the calibration results, it would be possible to detect the positional accuracy of the robot larger than 0.4 mm in the camera coordinate frame. However, after improving the robot-camera calibration matrix using an industrial camera with CCD sensors, it would be possible to detect the minor pose inaccuracies in the robot base coordinate frame.

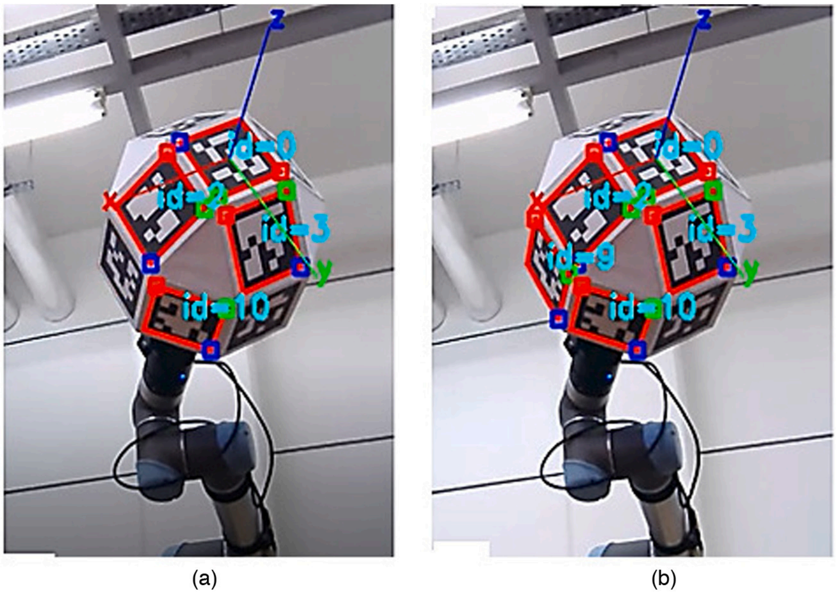


Fig. 19. The blinking-marker phenomena in the marker detection operation (a) maker#9 not being detected, and (b) marker#9 is detected.

6. Conclusion

A novel robot calibration framework based on computer vision and fiducial marker systems has been proposed in this paper. The main goal of this paper was to propose a calibration framework that minimizes the cost, complexity, and difficulty of the current standard calibration methods, particularly in the pose estimation step, which is a challenging part of robot calibration. The proposed calibration framework consisted of three stages: (i) Kinematic modeling of the robot using the POE formulas, (ii) pose estimation using the proposed vision-based system, and (iii) kinematic parameters identification.

The POE formulas successfully overcame the problem of singularity, which is the issue of the D-H based kinematic models. The proposed vision-based pose estimation system was designed based on a monocular vision system and a map of ArUco markers in eye-to-hand configuration. Such a breakthrough obviates the problems associated with standard calibration methods such as computational costs, occlusions, manual interventions, FOV, illumination, and system setup complexities. Simulation-based and real-world experiments proved that the accuracy of the proposed calibration framework is good; nevertheless, the accuracy of the system needs to be improved.

As a future work, our studies will focus on finding ways to improve the positional accuracy of the proposed calibration framework, particularly by minimizing error sources such as non-uniform illuminations, vertex jitter, and low-accurate hand–eye calibration matrix. Errors were mainly related to the quality of the camera with the CMOS sensor. Thus, high-quality industrial cameras with CCD sensors will be used to study the effect of cameras on the accuracy of the proposed calibration framework. Moreover, the proposed calibration framework would extend by adding the kinematic compensation step to the calibration procedure. Finally, in order to validate the performance of the proposed calibration framework, some real-world calibration operations will be performed on industrial robots working in manufacturing plants suffering positional inaccuracies, and LTS systems will be applied in parallel to assess the performance of the proposed calibration framework in practical applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Lin, H. Zhao, H. Ding, Spindle configuration analysis and optimization considering the deformation in robotic machining applications, Robot. Comput.-Integr. Manuf. 54 (June) (2018) 83–95, http://dx.doi.org/10.1016/j.rcim.2018.05.005.

[2] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, L.T. Tunc, Robots in machining, CIRP Ann. 68 (2) (2019) 799–822, http://dx.doi.org/10.1016/j.cirp.2019.05.009.

[3] ISO, ISO 9283:1998 Manipulating Industrial Robots — Performance Criteria and Related Test Methods, 2000, International Organization for Standardization, Geneva, CH, 1998.

[4] A. Nubiola, M. Slamani, A. Joubair, I.A. Bonev, Comparison of two calibration methods for a small industrial robot based on an optical CMM and a laser tracker, Robotica 32 (3) (2014) 447–466, http://dx.doi.org/10.1017/S0263574713000714.

[5] B. Karan, M. Vukobratović, Calibration and accuracy of manipulation robot models—An overview, Mech. Mach. Theory 29 (3) (1994) 479–500, http://dx.doi.org/10.1016/0094-114X(94)90130-9.

[6] M. Abderrahim, A. Khamis, S. Garrido, L. Moreno, Accuracy and calibration issues of industrial manipulators, Ind. Robot.: Progr. Simul. Appl. (2006) 131–146, http://dx.doi.org/10.5772/4895.

[7] Chen-Gang, Li-tong, Chu-Ming, J.Q. Xuan, S.H. Xu, Review on kinematics calibration technology of serial robots, Int. J. Precis. Eng. Manuf. 15 (8) (2014) 1759–1774, http://dx.doi.org/10.1007/s12541-014-0528-1.

[8] Wikipedia contributors, Calibration — Wikipedia the free encyclopedia, 2020, https://en.wikipedia.org/w/index.php?title=Calibration&oldid=982670069.

[9] R. Schmitt, M. Peterek, E. Morse, W. Knapp, M. Galetto, F. Härtig, G. Goch, B. Hughes, A. Forbes, W. Estler, Advances in large-scale metrology – Review and future trends, CIRP Ann. 65 (2) (2016) 643–665, http://dx.doi.org/10.1016/j.cirp.2016.05.002.

[10] I.A. Sultan, J.G. Wager, Simplified theodolite calibration for robot metrology, Adv. Robot. 16 (7) (2002) 653–671, http://dx.doi.org/10.1163/156855302260390764.

[11] D. Whitney, C. Lozinski, J. Rourke, Industrial robot forward calibration method and results, J. Dyn. Syst. Meas. Control 108 (1) (1986) 1–8, http://dx.doi.org/10.1115/1.3143737.

[12] U.S.P. Morris R. Driels, Robot calibration using an automatic theodolite, Int. J. Adv. Manuf. Technol. 9 (1994) 114–125, http://dx.doi.org/10.1007/BF01750418.

[13] W. Veitschegger, C. haur Wu, A method for calibrating and compensating robot kinematic errors, in: Proceedings. 1987 IEEE International Conference on Robotics and Automation, vol. 4, 1987, pp. 39–44, http://dx.doi.org/10.1109/ROBOT.1987.1087839.

[14] B. Mooring, S. Padavala, The effect of kinematic model complexity on manipulator accuracy, in: Proceedings, 1989 International Conference on Robotics and Automation, IEEE, 1989, pp. 593–598, http://dx.doi.org/10.1109/ROBOT.1989.100049.

[15] S. Gharaaty, T. Shu, A. Joubair, W.F. Xie, I.A. Bonev, Online pose correction of an industrial robot using an optical coordinate measure machine system, Int. J. Adv. Robot. Syst. 15 (4) (2018) 1729881418787915, http://dx.doi.org/10.1177/1729881418787915.

[16] A. Nubiola, I.A. Bonev, Absolute calibration of an ABB IRB 1600 robot using a laser tracker, Robot. Comput.-Integr. Manuf. 29 (1) (2013) 236–245, http://dx.doi.org/10.1016/j.rcim.2012.06.004.

[17] Y. Bai, H. Zhuang, Z.S. Roth, Experiment study of PUMA robot calibration using a laser tracking system, in: Proceedings of the 2003 IEEE International Workshop on Soft Computing in Industrial Applications, 2003. SMCia/03, 2003, pp. 139–144, http://dx.doi.org/10.1109/SMCIA.2003.1231359.

[18] G. Alici, B. Shirinzadeh, A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing, Mech. Mach. Theory 40 (8) (2005) 879–906, http://dx.doi.org/10.1016/j.mechmachtheory.2004.12.012.

[19] S.H. Ye, Y. Wang, Y.J. Ren, D.K. Li, Robot calibration using iteration and differential kinematics, J. Phys. Conf. Ser. 48 (2006) 1–6, http://dx.doi.org/10.1088/1742-6596/48/1/001.

[20] B. Mooring, Z. Roth, M. Driels, Fundamentals of Manipulator Calibration, in: A Wiley-Interscience publication, Wiley, 1991.

[21] K. Lau, R. Hocken, W. Haight, Automatic laser tracking interferometer system for robot metrology, Precis. Eng. 8 (1) (1986) 3–8, http://dx.doi.org/10.1016/0141-6359(86)90002-4.

[22] R. Podoloff, W. Seering, B. Hunter, An accuracy test procedure for robotic manipulators utilizing a vision based, 3-D position sensing system, in: 1984 American Control Conference, 1984, pp. 19–22, http://dx.doi.org/10.23919/ACC.1984.4788348.

[23] T. Clarke, X. Wang, The control of a robot end-effector using photogrammetry, Int. Arch. Photogr. Remote Sensing 33 (2000) 137–142.

[24] M. Švaco, B. Šekoranja, F. Šuligoj, B. Jerbić, Calibration of an industrial robot using a stereo vision system, Procedia Eng. 69 (2014) 459–463, http://dx.doi.org/10.1016/j.proeng.2014.03.012.

[25] C. Möller, H.C. Schmidt, N.H. Shah, J. Wollnack, Enhanced absolute accuracy of an industrial milling robot using stereo camera system, in: 3rd International Conference on System-Integrated Intelligence: New Challenges for Product and Production Engineering, Proc. Technol. 26 (2016) 389–398, http://dx.doi.org/10.1016/j.protcy.2016.08.050.

[26] X. Zhang, Y. Song, Y. Yang, H. Pan, Stereo vision based autonomous robot calibration, Robot. Auton. Syst. 93 (2017) 43–51, http://dx.doi.org/10.1016/j.robot.2017.04.001.

[27] A. Filion, A. Joubair, A.S. Tahan, I.A. Bonev, Robot calibration using a portable photogrammetry system, Robot. Comput.-Integr. Manuf. 49 (2018) 77–87, http://dx.doi.org/10.1016/j.rcim.2017.05.004.

[28] G. van Albada, A. Visser, J. Lagerberg, L. Hertzberger, Publication, A Portable Measuring System for Robot Calibration, vol. 795, 1995, pp. 441–448.

[29] Y.-L. Kuo, B.-H. Liu, C.-Y. Wu, Pose determination of a robot manipulator based on monocular vision, IEEE Access 4 (2016) 8454–8464, http://dx.doi.org/10.1109/ACCESS.2016.2633378.

[30] J.M.S. Motta, G.C. de Carvalho, R. McMaster, Robot calibration using a 3D vision-based measurement system with a single camera, Robot. Comput.-Integr. Manuf. 17 (6) (2001) 487–497, http://dx.doi.org/10.1016/S0736-5845(01)00024-2.

[31] G. Du, P. Zhang, Online robot calibration based on vision measurement, Robot. Comput.-Integr. Manuf. 29 (6) (2013) 484–492, http://dx.doi.org/10.1016/j.rcim.2013.05.003.

[32] R. Wang, A. Wu, X. Chen, J. Wang, A point and distance constraint based 6R robot calibration method through machine vision, Robot. Comput.-Integr. Manuf. 65 (2020) 101959, http://dx.doi.org/10.1016/j.rcim.2020.101959.

[33] M.B. Alatise, G.P. Hancke, Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter, Sensors 17 (10) (2017) http://dx.doi.org/10.3390/s17102164.

[34] B. Liu, F. Zhang, X. Qu, A method for improving the pose accuracy of a robot manipulator based on multi-sensor combined measurement and data fusion, Sensors 15 (4) (2015) 7933–7952, http://dx.doi.org/10.3390/s150407933.

[35] G. Du, P. Zhang, D. Li, Online robot calibration based on hybrid sensors using Kalman filters, Robot. Comput.-Integr. Manuf. 31 (2015) 91–100, http://dx.doi.org/10.1016/j.rcim.2014.08.002.

[36] G. Du, H. Shao, Y. Chen, P. Zhang, X. Liu, An online method for serial robot self-calibration with CMAC and UKF, Robot. Comput.-Integr. Manuf. 42 (2016) 39–48, http://dx.doi.org/10.1016/j.rcim.2016.05.006.

[37] G. Freitas, A. Leite, F. Lizarralde, Kinematic control of constrained robotic systems, Sba Controle Autom. Soc. Bras. Autom. 22 (2011) 559–572, http://dx.doi.org/10.1590/S0103-17592011000600002.

[38] A. Gupta, S. Arora, Industrial Automation and Robotics, University Science Press, 2009.

[39] H. Zhuang, Z.S. Roth, Camera-Aided Robot Calibration, CRC Press, 1996.

[40] G. Gatti, G. Danieli, A practical approach to compensate for geometric errors in measuring arms: application to a six-degree-of-freedom kinematic structure, Meas. Sci. Technol. 19 (1) (2007) 015107, http://dx.doi.org/10.1088/0957-0233/19/1/015107.

[41] K. Schröer, S.L. Albright, M. Grethlein, Complete, minimal and model-continuous kinematic models for robot calibration, Robot. Comput.-Integr. Manuf. 13 (1) (1997) 73–85, http://dx.doi.org/10.1016/S0736-5845(96)00025-7.

[42] S. Hayati, Robot arm geometric link parameter estimation, in: The 22nd IEEE Conference on Decision and Control, 1983, pp. 1477–1483, http://dx.doi.org/10.1109/CDC.1983.269783.

[43] S. Hayati, M. Mirmirani, Improving the absolute positioning accuracy of robot manipulators, J. Robot. Syst. 2 (4) (1985) 397–413, http://dx.doi.org/10.1002/rob.4620020406.

[44] H. Stone, A. Sanderson, A prototype arm signature identification system, in: Proceedings. 1987 IEEE International Conference on Robotics and Automation, 4, 1987, pp. 175–182, http://dx.doi.org/10.1109/ROBOT.1987.1087835.

[45] F.C. Park, K. Okamura, Kinematic calibration and the product of exponentials formula, in: Advances in Robot Kinematics and Computational Geometry, Springer, 1994, pp. 119–128, http://dx.doi.org/10.1007/978-94-015-8348-0-12.

[46] R. Brockett, Robotic manipulators and the product of exponential formulas, in: P.A. Fuhrmann (Ed.), Mathematical Theory of Networks and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 1984, pp. 120–129, http://dx.doi.org/10.1007/BFb0031048.

[47] R. He, Y. Zhao, S. Yang, S. Yang, Kinematic-parameter identification for serial-robot calibration based on POE formula, IEEE Trans. Robot. 26 (3) (2010) 411–423, http://dx.doi.org/10.1109/TRO.2010.2047529.

[48] R.M. Murray, Z. Li, S.S. Sastry, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.

[49] K. Okamura, F. Park, Kinematic calibration using the product of exponentials formula, Robotica 14 (4) (1996) 415–421, http://dx.doi.org/10.1017/S0263574700019810.

[50] M. Meggiolaro, S. Dubowsky, An analytical method to eliminate the redundant parameters in robot calibration, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 4, 2000, pp. 3609–3615 vol.4, http://dx.doi.org/10.1109/ROBOT.2000.845294.

[51] W. Khalil, M. Gautier, C. Enguehard, Identifiable parameters and optimum configurations for robots calibration, Robotica 9 (1) (1991) 63–70, http://dx.doi.org/10.1017/S0263574700015575.

[52] I.-M. Chen, G. Yang, Kinematic calibration of modular re-configurable robots using product-of-exponential formula, J. Robot. Syst. 14 (11) (1997) 807–821, http://dx.doi.org/10.1109/ROBOT.1997.606775.

[53] S. Garrido-Jurado, R.M. noz Salinas, F. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using mixed integer linear programming, Pattern Recognit. 51 (2016) 481–491, http://dx.doi.org/10.1016/j.patcog.2015.09.023.

[54] R.M. noz-Salinas, M.J. Marín-Jiménez, E. Yeguas-Bolivar, R.M. Carnicer, Mapping and localization from planar markers, Pattern Recognit. 73 (2016) 158–171, http://dx.doi.org/10.1016/j.patcog.2017.08.010.

[55] H. Sarmadi, R. Muñoz Salinas, M. Berbís, R. Medina-Carnicer, Simultaneous multi-view camera pose estimation and object tracking with squared planar markers, IEEE Access 7 (2019) 22927–22940, http://dx.doi.org/10.1109/ACCESS.2019.2896648.

[56] A. Tsoukalas, A. Tzes, F. Khorrami, Relative pose estimation of unmanned aerial systems, in: 2018 26th Mediterranean Conference on Control and Automation (MED), 2018, pp. 155–160, http://dx.doi.org/10.1109/MED.2018.8442959.

[57] The OpenCV Reference Manual [Electronic Resource], third ed., 2019, http://opencv.org/.

[58] S. Garrido-Jurado, R.M. noz Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognit. 47 (6) (2014) 2280–2292, http://dx.doi.org/10.1016/j.patcog.2014.01.005.

[59] E. Davies, Computer and Machine Vision : Theory, Algorithms, Practicalities, fourth ed., Academic Press, 2012.

[60] MVTec Software GmbH, HALCON:a product of MVtec, Solution Guide III-C 3D vision, 2020.

[61] I.-M. Chen, G. Yang, C. Tan, H. Song, Yeo, Local POE model for robot kinematic calibration, Mech. Mach. Theory 36 (2001) 1215–1239, http://dx.doi.org/10.1016/S0094-114X(01)00048-9.

[62] F.J. Romero-Ramirez, R. Muñoz Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers, Image Vis. Comput. 76 (2018) 38–47, http://dx.doi.org/10.1016/j.imavis.2018.05.004.

[63] M. Fiala, ARTag, a fiducial marker system using digital techniques, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, 2005, pp. 590–596, http://dx.doi.org/10.1109/CVPR.2005.74.

[64] J. Liu, J. Wu, X. Li, Robust and accurate hand–eye calibration method based on schur matric decomposition, Sensors 19 (20) (2019) http://dx.doi.org/10.3390/s19204490.