Advanced Robotics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Vision-aided precise positioning for long-reach robotic manipulators using local calibration

Petri Mäkinen, Pauli Mustalahti, Sirpa Launis & Jouni Mattila

To cite this article: Petri Mäkinen, Pauli Mustalahti, Sirpa Launis & Jouni Mattila (30 Dec 2023): Vision-aided precise positioning for long-reach robotic manipulators using local calibration, Advanced Robotics, DOI: 10.1080/01691864.2023.2299852

To link to this article: https://doi.org/10.1080/01691864.2023.2299852

ADVANCED ROBOTICS
https://doi.org/10.1080/01691864.2023.2299852

FULL PAPER

OPEN ACCESS

Vision-aided precise positioning for long-reach robotic manipulators using local calibration

Petri Mäkinen[a], Pauli Mustalahti[a], Sirpa Launis[b] and Jouni Mattila[a]

[a]Automation Technology and Mechanical Engineering, Tampere University, Tampere, Finland; [b]Sandvik Mining and Rock Technology, Tampere, Finland

ABSTRACT
A vision-based guidance methodology is proposed for precise positioning of the tool center point (TCP) of heavy-duty, long-reach (HDLR) manipulators. HDLR manipulators are non-rigid structures with many nonlinearities. Therefore, conventional rigid-body–based modeling and control methods issue challenges for accurate TCP positioning. To compensate for these errors, we compute the pose error between the TCP and an object of interest (OOI) directly in the camera frame, while using motion-based local calibration to find the extrinsic sensor-to-robot correspondence. The proposed pipeline for local calibration is twofold: first, the detected tool is oriented perpendicularly with respect to the OOI. Second, range adjustment is performed in the local planar plane by exploiting the visual measurements. Two methods for adjusting the range were examined: a line equation–based method and a trajectory matching–based method. Real-time experiments were conducted using a HDLR manipulator with a 5 m reach, and visual fiducial markers were used as detectable objects for the visual sensor. The experimental results demonstrated that the proposed methodology can provide sub-centimeter positioning accuracy, which is very challenging to achieve with HDLR manipulators due to their characteristic uncertainties.

ARTICLE HISTORY
Received 1 June 2023
Revised 12 September 2023
Accepted 9 December 2023

KEYWORDS
Vision-based control; precise positioning; eye-in-hand; local calibration; heavy-duty manipulators

1. Introduction

Visual sensors, such as cameras and laser scanners, are becoming an essential part of robotic systems. Traditionally, robots' work tasks have been pre-programed by utilizing precise joint sensors with high repeatability [1]. However, emerging intelligent algorithms combined with visual detection and pose (3 DOF position and 3 DOF orientation) estimation provide a way for autonomous robotic systems performing work tasks based on non-contact visual sensing [2]. A typical scenario is visual servoing, in which an object of interest (OOI) is detected using a visual sensor, and then the tool center point (TCP) of the manipulator is guided toward the OOI based on the visual feedback [3, 4]. Methods of visual servoing are generally classified into position-based (PBVS) [5], image-based (IBVS) [6], and hybrid systems [7]. In a PBVS system, the control error is defined in Cartesian 3D coordinates, and the control algorithm utilizes the robot's kinematic model along with camera calibration parameters. In an IBVS system, the control law is defined in the 2D image plane using image features directly. The relationship between the image plane and the robot is established using an image Jacobian matrix, which describes a nonlinear mapping between the image feature errors and the pose of the robot [8]. Hybrid systems attempt to utilize the advantages of both the PBVS and IBVS. Visual control can also be realized in an open-loop manner, where the continuous feedback loop is omitted. Applications for vision-based control include grasping, surgical instruments, insertions, pick and place, and so forth.

The first of the two main causes of inaccuracies in a robotic manipulator's TCP positioning results from incorrect variables in kinematic modeling, such as the Denavit-Hartenberg (DH) parameters. The second cause are non-kinematic errors caused by, for example, structural bending and flexibility, thermal effects, backlash, and sensor resolution [9]. The problem of precise TCP positioning in robotics is highlighted in the context of heavy-duty, long-reach (HDLR) manipulators that are utilized in mobile machines. For economic reasons, it is desirable that HDLR manipulators are constructed to be as lightweight as possible while able to handle significant load masses. Such conditions result in high structural flexibility and other uncertainties that are not an issue with small and compact industrial robots. For HDLR manipulators, traditional rigid-body kinematics-based control is not sufficient for precise TCP positioning due to the non-rigid structures. However, precise

CONTACT Petri Mäkinen petri.makinen@tuni.fi

positioning is an essential requirement for automated work tasks even with a human-in-the-loop and eventually fully autonomous machines. Therefore, HDLR manipulators could benefit considerably from camera-based guidance that is utilized to minimize the positioning error between the TCP and an OOI, which thus bypasses the weaknesses of the imprecise yet computationally efficient rigid-body–based kinematics, along with non-kinematic error sources.

Attaching a camera near the tip of a robotic manipulator is known as an eye-in-hand configuration [10]. The rigid relation between the camera frame (eye) and the TCP of the manipulator (hand) is described with a transformation matrix estimated using extrinsic calibration [11]. This static mapping allows information obtained using the camera, or another visual sensor, to be expressed in the robot's coordinate system. Typically, the extrinsic calibration procedure requires a predefined object with a set world frame, such as a checkerboard or a circular grid [12–15], which is used to estimate the hand-eye transformation. The procedure also requires taking images from different distances and angles with respect to the calibration object. Consequently, this setup is mostly feasible for structured environments in controlled factory settings. A key differentiation between mobile machines with on-board HDLR manipulators and conventional industrial robots is that the former operate in dynamic, unstructured environments. Thus, using a predefined calibration object to estimate the hand-eye transform in mines, fields, or plantations, is not realistic or practical.

Vision-based control in mining applications, including the classic peg-in-hole problem, was discussed in [16–18]. In the peg-in-hole problem, the positioning of the tool to a desired OOI is essentially reduced to a planar positioning problem in the vicinity of the OOI, which is also exploited in our methodology. Visual servoing has been widely employed to solve the peg-in-hole problem in repetitive assembly tasks related to factory automation [19]. Thus, these methods have mostly focused on structured scenes [20], whereas HDLR manipulators work in dynamic and unstructured environments. Other mining-related studies with visual sensors include [21], in which a laser scanner was utilized for collision avoidance. Studies utilizing laser scanners in mines also include [22, 23]. In [24], an eye-to-hand configuration was used for position-based visual guidance of a heavy-duty rock-breaking manipulator. Specific markers for calibration purposes were distributed into the workspace and a considerable number of measurements were conducted to estimate the extrinsic camera-to-robot calibration parameters. In [25], PBVS using an RGB-D camera was investigated for a multi-joint hydraulic manipulator, but the camera was rigidly mounted at a known location along the manipulator's kinematic chain, and no explicit hand-eye calibration was described. In [26], sub-centimeter absolute positioning accuracy was achieved with a HDLR manipulator by using a total station network. This method requires a large space in which to operate and a considerable investment in the sensors.

In this paper, the objective was to drive the tool of a HDLR manipulator to an OOI using a low-cost camera. HDLR manipulators would benefit from vision-based guidance systems especially in auxiliary tasks requiring precise (sub-centimeter) TCP positioning accuracy. Potential tasks for increased levels of automation include, for example, swapping tools or navigating the TCP to a pre-drilled hole. The desired positioning accuracy in this type of application is $+/-5$ mm, which is very difficult to achieve without a highly skilled human operator due to the characteristic uncertainties of HDLR manipulators. To solve this problem, we propose computing the pose error between the tool and an OOI directly in the camera frame (which relates our approach to PBVS), while using motion-based local calibration to align the camera frame with the TCP frame. Notably, the procedure does not require placing dedicated calibration objects in the environment. The pipeline for motion-based local calibration is twofold: First, the tool's torsion axis is oriented perpendicularly w.r.t. the OOI. Then, range adjustment is performed in the local (planar) plane while the perpendicular configuration is maintained. Two methods for adjusting the range are presented: a line equation–based method and a trajectory matching–based method. The first method is simple and easy to implement, while the second method is more specific and complex to implement. Real-time experiments were conducted on a HDLR manipulator with a 5 m reach, and a low-cost stereo camera was attached near the tip. In the experiments we used ArUco markers [27] as 'generic' representations of a tool and an OOI. The long-term goal is to use real-world objects. A key assumption of our methodology is that the tool and the OOI are visible to the camera, they can be detected, and their 6 DOF poses can be estimated. As the main contribution, it is shown that the image-based positioning error between the tool and the OOI can be reduced to the sub-centimeter range using the proposed methods, which is very challenging to achieve with HDLR manipulators due to their characteristics.

The rest of the paper is organized as follows. Robotics and control-related preliminaries are given in Section 2, the methods used are detailed in Section 3, the experimental setup is described in Section 4, the results are discussed in Section 5, and finally, the paper is concluded in Section 6.

2. Robotics and control preliminaries

The pose $\mathbf{x} \in \mathbb{R}^6$ of a robotic manipulator in its operational space describes the position and orientation of the TCP frame w.r.t. the base frame of the manipulator. The transition between the operational space and the joint space can be written using forward kinematic equations when individual joint variables $\mathbf{q} \in \mathbb{R}^n$ are known:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\theta} \end{bmatrix} = \mathbf{f}(\mathbf{q}), \quad \mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ represents the TCP position and $\boldsymbol{\theta} \in \mathbb{R}^3$ its orientation. The rotation of the TCP frame is expressed using minimal representation (i.e. Euler angles). The relation between the TCP's linear and angular velocities and the joint velocities is computed as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{2}$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix describing the linear mapping from joint space velocities $\dot{\mathbf{q}} \in \mathbb{R}^n$ to the operational space velocities $\dot{\mathbf{x}} \in \mathbb{R}^6$. If the TCP velocities are known, the joint velocities can be obtained by using the inverse Jacobian as follows:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}. \tag{3}$$

Assuming that the desired TCP position $\mathbf{p}_d$ and the orientation $\boldsymbol{\theta}_d$ are known, along with the respective desired linear velocities $\dot{\mathbf{p}}_d$ and angular velocities $\dot{\boldsymbol{\theta}}_d$, the desired joint velocities $\dot{\mathbf{q}}_d$ can be computed using Equation (3) as follows:

$$\dot{\mathbf{q}}_d = \mathbf{J}^{-1}(\mathbf{q}) \begin{bmatrix} \dot{\mathbf{p}}_d + \mathbf{K}_p(\mathbf{p}_d - \mathbf{p}) \\ \dot{\boldsymbol{\theta}}_d + \mathbf{K}_\theta \delta\mathbf{r} \end{bmatrix}, \tag{4}$$

where $\mathbf{K}_p$ and $\mathbf{K}_\theta$ are the control gains associated with the incorporated position and orientation feedback. Furthermore, $\delta\mathbf{r}$ denotes the orientation error, expressed in terms of quaternion. The desired joint positions are then integrated from the desired joint velocities:

$$\mathbf{q}_d = \int \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}_d \,\mathrm{d}t. \tag{5}$$

The control input vector $\mathbf{u}$ is then formulated as follows:

$$\mathbf{u} = \mathbf{K}_v(\mathbf{q}_d - \mathbf{q}), \tag{6}$$

where $\mathbf{K}_v$ is a matrix containing the joint control gains.

3. Methods

The overall procedure for guiding the tool to an OOI using local calibration is to first orient the tool's torsion axis perpendicularly w.r.t. the OOI based on the visual measurements. Then, motion-based range adjustment is conducted in the local (planar) plane while the perpendicular configuration is maintained. Two methods for adjusting the range were examined: a line equation–based method and a trajectory matching-based method.

The line equation–based method utilizes a circular path and a line equation to compute the desired TCP position. The OOI is required to be visible to the camera at all times. This method relies on the uniformity of the camera frame and the TCP frame, so that the plane of motion is planar in both frames. The depth parameter is not directly considered.

In contrast, the trajectory matching–based method employs visual odometry/simultaneous localization and mapping (VO/SLAM). VO/SLAM-based TCP pose tracking of a HDLR manipulator in a confined space was studied in [28]. Thus, this method does not require the OOI to be visible during the calibration. This method is also more robust in the camera placement, as the camera's rotation w.r.t. the manipulator's base frame is included in the computed calibration matrix. Furthermore, the depth parameter is considered although the calibration path is planar.

The objective of the local calibration procedure is to obtain a reference pose $\mathbf{x}_{ref} \in \mathbb{R}^6$ for the TCP, while the perpendicularity is maintained. Then, using the basic rigid-body–based kinematics and control methods presented in Section 2, the tool is driven to the desired OOI based on the positioning error measured directly from the image. The methodologies for adjusting the orientation and the range are detailed below.

3.1. Orientation adjustment

In the case of the ArUco markers, the goal was to orient the tool marker to match the OOI's orientation so that the last joint's torsion axis was perpendicular w.r.t. to the OOI. The poses of the OOI frame $^{\mathbf{C}}\mathbf{T}_{\mathbf{O}}$ and the tool frame $^{\mathbf{C}}\mathbf{T}_{\mathbf{T}}$ are obtained from image-based computations. The rotation components are formulated as unit quaternions, $\mathbf{r}_O$ and $\mathbf{r}_T$. Then, the rotation difference $\mathbf{r}_\Delta$ between the frames is computed as the product of the two quaternions, while inversing the tool frame quaternion:

$$\mathbf{r}_\Delta = \mathbf{r}_O\mathbf{r}_T^{-1}. \tag{7}$$

The last joint's torsion axis is oriented in a perpendicular configuration w.r.t. the OOI frame by computing the manipulator's reference TCP pose $\mathbf{x}_{ref}$ in the form of a transformation matrix as follows:

$$\mathbf{T}_{\text{ref}} = {}^{\mathbf{B}}\mathbf{T}_{\mathbf{T}} \begin{bmatrix} \mathbf{R}_{\Delta} & \mathbf{0}^{\text{T}} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{8}$$

where $\mathbf{R}_{\Delta} \in \mathbb{R}^{3\times3}$ denotes the rotation matrix computed from the quaternion product in Equation (7), and $\mathbf{0} = [0,\ 0,\ 0]$.

3.2. Range adjustment

3.2.1. Line equation–based method

In accordance with [29], the transformation matrix relating the base of the manipulator to the TCP frame comprises unit orientation vectors ($\mathbf{n}$, $\mathbf{s}$, $\mathbf{a}$) and the position vector $\mathbf{p}$:

$${}^{\mathbf{B}}\mathbf{T}_{\mathbf{T}}(\mathbf{q}) = \begin{bmatrix} {}^{\mathbf{B}}\mathbf{n}_{\mathbf{T}}(\mathbf{q}) & {}^{\mathbf{B}}\mathbf{s}_{\mathbf{T}}(\mathbf{q}) & {}^{\mathbf{B}}\mathbf{a}_{\mathbf{T}}(\mathbf{q}) & {}^{\mathbf{B}}\mathbf{p}_{\mathbf{T}}(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

As discussed, the control problem in the OOI's vicinity is treated as a planar positioning problem. Thus, the local 2D plane for the range adjustment is defined by the unit orientation vectors by using the current pose of the manipulator resulting from the orientation adjustment. For the line equation–based method, a circular path is designed using point-to-point motion in Cartesian 3D space as follows:

$$x(\phi) = p_1 + r\cos(\phi)n_1 + r\sin(\phi)s_1, \tag{10}$$

$$y(\phi) = p_2 + r\cos(\phi)n_2 + r\sin(\phi)s_2, \tag{11}$$

$$z(\phi) = p_3 + r\cos(\phi)n_3 + r\sin(\phi)s_3, \tag{12}$$

where r denotes the circle's radius, $\mathbf{p}_{init} = [p_1,\ p_2,\ p_3]^T$ denotes the initial TCP position, the unit orientation vectors $\mathbf{n} \in \mathbb{R}^3$ and $\mathbf{s} \in \mathbb{R}^3$ are extracted from Equation (9), and $\phi \in [0, 2\pi]$. The first and last points of the path are set to the center of the designed circular path.

The path is executed while holding the current TCP orientation. During the path execution, the image-based distance error between the tool and the OOI is maintained in metric form:

$$d_{\epsilon} = \sqrt{({}^{O}p_x - {}^{T}p_x)^2 + ({}^{O}p_y - {}^{T}p_y)^2}, \tag{13}$$

where p_x and p_y denote the metric positions of the detected tool and the OOI measured in the camera frame, respectively. After the path is completed, the minimum distance error $d_{\epsilon,\text{min}}$ is obtained. The algorithm saves the initial pose of the manipulator (the circle's center) and the pose corresponding to the measured minimum distance error:

$$\mathbf{x}_{init} = \begin{bmatrix} \mathbf{p}_{init} \\ \boldsymbol{\theta}_{init} \end{bmatrix}, \quad \mathbf{x}_{\epsilon} = \begin{bmatrix} \mathbf{p}_{\epsilon} \\ \boldsymbol{\theta}_{\epsilon} \end{bmatrix}. \tag{14}$$

The normalized unit vector pointing toward the OOI is then computed using the two points as follows:

$$\mathbf{p}_u = \frac{\mathbf{p}_{\epsilon} - \mathbf{p}_{init}}{\|\mathbf{p}_{\epsilon} - \mathbf{p}_{init}\|}. \tag{15}$$

The reference TCP pose $\mathbf{x}_{ref} \in \mathbb{R}^6$ is expressed using the following transformation matrix:

$$\mathbf{T}_{\text{ref}} = \begin{bmatrix} {}^{\mathbf{B}}\mathbf{R}_{\mathbf{T}} & \mathbf{p}_{init} + \mathbf{p}_u(r + d_{\epsilon,\text{min}}) \\ \mathbf{0} & 1 \end{bmatrix}, \tag{16}$$

where the current TCP orientation is held. The reference position is computed with the line equation using the initial position (the circle's center), the deduced directional unit vector, the circle's radius, and the minimum distance error between the tool and the OOI. The circular path and related variables are illustrated in Figure 1.

3.2.2. Trajectory matching–based method

Trajectory matching was explored as an alternative method for adjusting the range. This method is more complex but does not require detection of the OOI during the motion-based calibration. It is also not dependent on the orientation adjustment, although the perpendicular configuration is helpful when computing the pose error between the tool and an OOI.

First, a point-to-point calibration path in planar space is designed using the following equations:

$$x_{i+1} = x_i + L_i\cos(\xi)n_1 + L_i\sin(\xi)s_1, \tag{17}$$

$$y_{i+1} = y_i + L_i\cos(\xi)n_2 + L_i\sin(\xi)s_2, \tag{18}$$

$$z_{i+1} = z_i + L_i\cos(\xi)n_3 + L_i\sin(\xi)s_3, \tag{19}$$

where $[x_1, y_1, z_1]^{\text{T}} = {}^{\mathbf{B}}\mathbf{p}_{\mathbf{T}}$, the unit orientation vectors $\mathbf{n} \in \mathbb{R}^3$ and $\mathbf{s} \in \mathbb{R}^3$ are obtained from Equation (9), the angle ξ is altered between 0 and $\frac{\pi}{4}$, and L_i denotes the length of the ith path segment. The path was designed to be asymmetric to improve the matching outcome.

The asymmetric path is executed, and two pose trajectories are saved: (i) The camera's pose trajectory is estimated using a VO/SLAM algorithm, and (ii) the TCP pose trajectory based on the kinematics model and the joint sensors is also obtained. The camera-to-kinematic model transformation matrix is then obtained using robust point set matching. The overall formalism is detailed in [30], which has two steps: (1) Coarse frame alignment is used to roughly orient the two point sets containing the pose trajectories to avoid an incorrect (e.g. mirrored) matching result and (2) fine matching

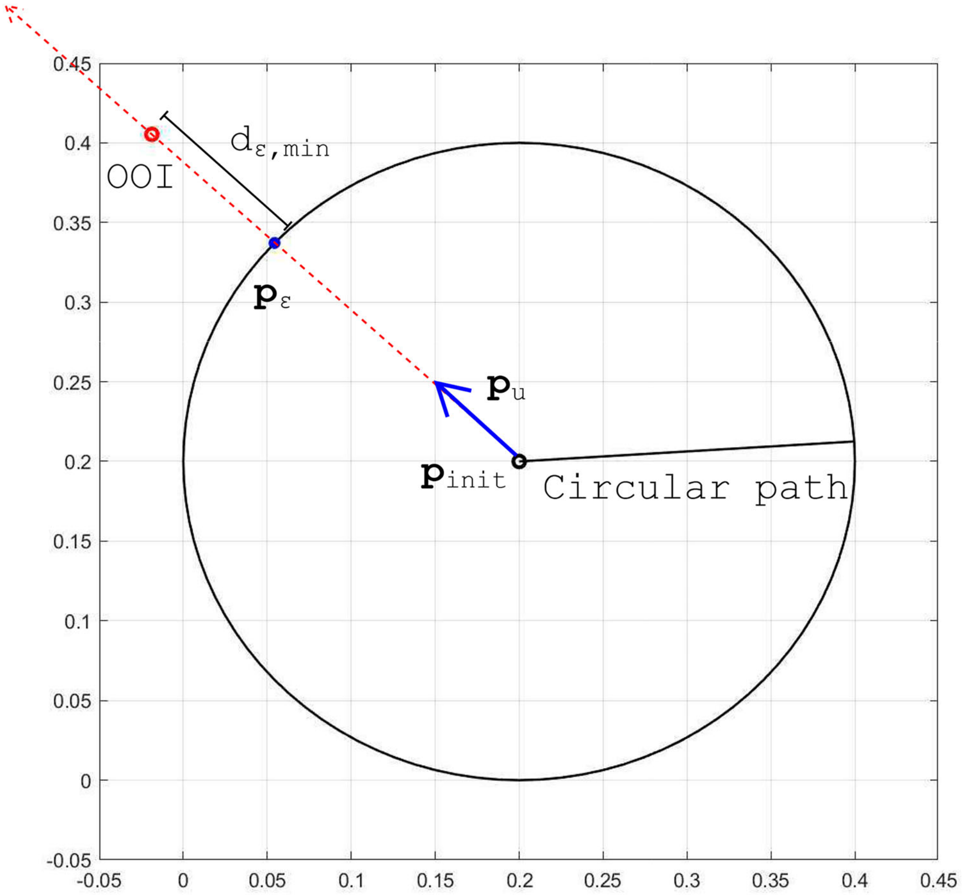

Figure 1. The principle of the line equation–based range adjustment.

using probabilistic hybrid mixture model-based point set matching [31]. This method utilizes the full 6 DOF pose data in the trajectory matching procedure, whereas most algorithms utilize only the 3 DOF position data. However, other accurate point set matching methods should also suffice.

After the point set matching sequence is completed, information measured in the camera frame can be expressed w.r.t. the manipulator's base frame. The locally calibrated position error between the tool and the OOI is written as follows:

$$\mathbf{T}_{\mathrm{lc}} = \begin{bmatrix} \mathbf{R}_{\mathrm{fm}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathrm{cfa}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & 1 \end{bmatrix} {}^{\mathrm{C}}\mathbf{T}_{\Delta}, \tag{20}$$

where $\mathbf{R}_{\mathrm{fm}}$ denotes the rotation matrix related to the fine matching, and $\mathbf{R}_{\mathrm{cfa}}$ denotes the rotation matrix related to the coarse frame alignment. The transformation matrix ${}^{\mathrm{C}}\mathbf{T}_{\Delta}$ contains the positioning error between the OOI and the tool frame, measured in the camera frame, as follows:

$${}^{\mathrm{C}}\mathbf{T}_{\Delta} = \begin{bmatrix} 1 & 0 & 0 & {}^{O}p_x - {}^{T}p_x \\ 0 & 1 & 0 & {}^{O}p_y - {}^{T}p_y \\ 0 & 0 & 1 & {}^{O}p_z - {}^{T}p_z - b \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

where b is a small offset to avoid collision between the tool and the OOI. Then, the reference TCP pose $\mathbf{x}_{ref} \in \mathbb{R}^6$ is obtained using the following transformation matrix:

$$\mathbf{T}_{\mathrm{ref}} = \begin{bmatrix} {}^{\mathrm{B}}\mathbf{R}_{\mathrm{T}} & \mathbf{p}_{\mathrm{lc}} + {}^{\mathrm{B}}\mathbf{p}_{\mathrm{T}} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{22}$$

where the tool orientation is held, and the locally calibrated image-based positioning error $\mathbf{p}_{\mathrm{lc}} \in \mathbb{R}^3$ is incorporated into the current position. While maintaining the local calibration, the reference TCP pose can be updated using the visual sensor and Equation (22) as often as required (within the limitations of the sensor's refresh rate). The principle for the trajectory matching–based method is illustrated in Figure 2.

4. Experimental setup

4.1. Vision system

ArUco marker detection and pose estimation were realized using OpenCV functions, which employ monocular methods. The metric scale was obtained based on the known marker size. The experimental setup is illustrated in Figure 3. A single marker representing the tool was

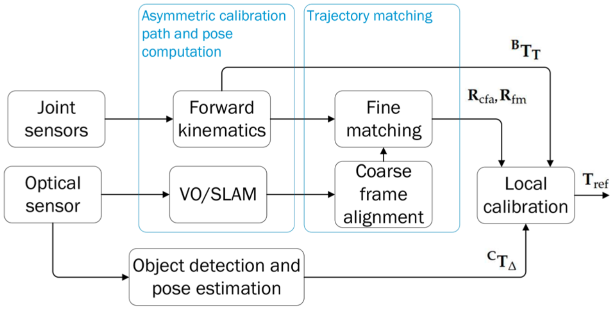

Figure 2. The pipeline for the trajectory matching–based range adjustment.

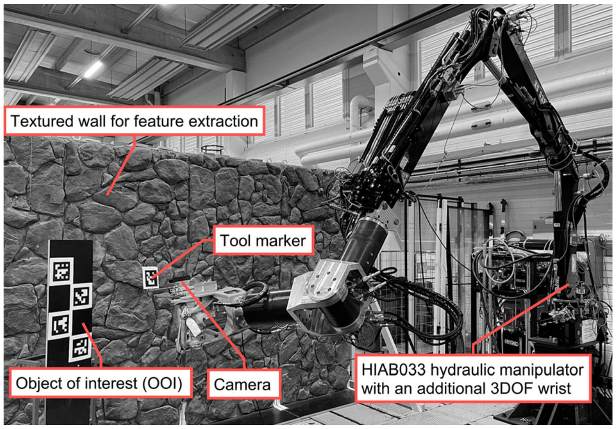
Textured wall for feature extraction
Tool marker
Object of interest (OOI)
Camera
HIAB033 hydraulic manipulator with an additional 3DOF wrist

Figure 3. The experimental setup. The OOI was in the middle of the three ArUco markers, and the marker attached at the end of the HDLR manipulator represented a detectable tool.

attached parallel to the torsion axis of the last joint of the manipulator. Additionally, the marker was set at approximately a 90° angle w.r.t. to the axis. A three-marker setup was used to construct the OOI. Specifically, the OOI was in the middle of the three markers, and the OOI detection algorithm was designed so that one detected marker out of the three is sufficient to deduce the pose of the OOI. The objective was to drive the tool marker to the OOI, with a depth offset to avoid collision.

A low-cost ZED2 stereo camera was attached near the TCP and parallel to the torsion axis of the last joint. The left lens was used to detect the marker and estimate the pose. Therefore, the pose error between the tool marker and the OOI was known in the camera frame. As for the VO/SLAM utilized in Subsection 3.2.2, the open-source ORB-SLAM2 stereo algorithm [32] was employed. An adequate surface for the VO/SLAM algorithm's feature extraction was provided by a textured wall. A PC running ROS was used to handle the vision system, and the required information was sent to the manipulator's Beckhoff control system via UDP. The camera settings were set to 720p at 60 FPS, and it was assumed that the intrinsic parameters were known.

Table 1. DH parameters of the HIAB033 with a 3 DOF wrist.

Joint	α_i	a_i	θ_i	d_i
rotation	$\pi/2$	a_1	θ_1	d_1
lift	0	a_2	θ_2	0
tilt	$\pi/2$	a_3	θ_3	d_3
wrist 1	$\pi/2$	0	θ_4	d_4
wrist 2	$-\pi/2$	0	θ_5	0
wrist 3	$\pi/2$	0	θ_6	d_6
link 7	$\pi/2$	0	0	0

4.2. Hydraulic manipulator with a 3 DOF wrist

A laboratory-installed HDLR manipulator with an approximately 5 m reach was used in the experiments. Using the DH convention, the forward kinematic model of the manipulator is formulated. The symbolic DH parameters are shown in Table 1, where 'Link 7' is a mock frame to unify the last frame with the tool marker's frame.

The rigid transformation matrix from the base of the manipulator to the TCP is computed as follows:

$$^{\mathbf{B}}\mathbf{T}_{\mathbf{T}} = \prod_{i=1}^{7} \mathbf{T}_i, \tag{23}$$

where $\mathbf{T}_i$ denotes the joint-specific transformation matrices, formulated as follows:

$$\mathbf{T}_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{24}$$

while using the respective DH parameters of each joint. Additionally, $s = sin$, and $c = cos$.

After the desired pose $\mathbf{x}_{ref}$ for the TCP was obtained using Equations (8), (16), or (22), point-to-point paths were generated using a quintic polynomial [33]. Open-loop visual control (looking and then moving) was utilized, as for HDLR manipulators in dynamic environments, safety is a significant concern and cameras are subject to robustness and reliability issues in such conditions. Instead, only the joint controllers work in closed-loop. Closed-loop visual control (looking and moving) is also possible by continuous usage of the camera feed, but requires constant and reliable vision of the tool and the OOI. By performing several 'looking and then moving' steps, the open-loop method can provide the same final positioning accuracy as the closed-loop method. The control system is described in Figure 4. For joint control, Equation (6) was utilized. For the two joints contributing the most (lift and tilt), PT-1 control was used, for which the transfer function is written as follows:

$$G(s) = \frac{K_p}{\tau s + 1}, \tag{25}$$

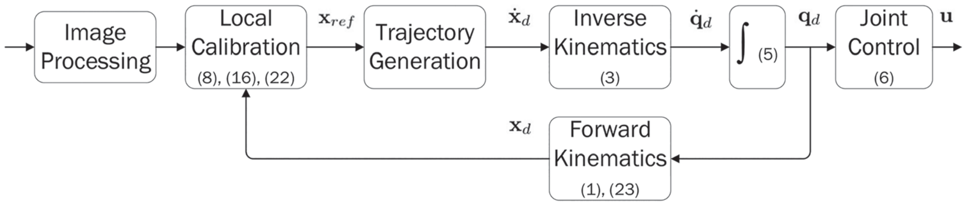

Figure 4. The control loop for the image-based reference pose. Image processing is conducted on a dedicated PC running ROS, whereas the rest of the process is handled on the Beckhoff control system. The numbers refer to the main related equations.

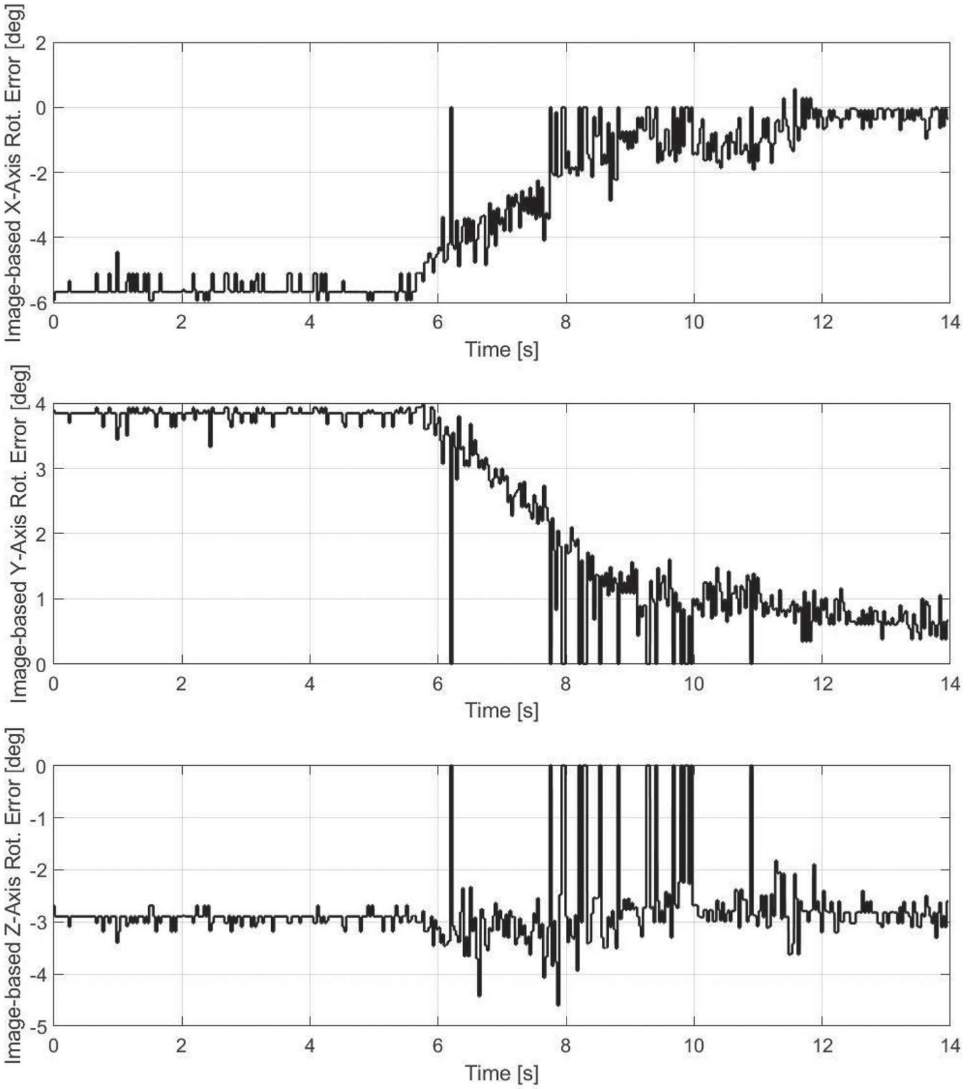

Figure 5. Image-based orientation errors between the tool marker and the OOI before and after the orientation adjustment, expressed in Euler XYZ angles. The control input was given at approximately 5.6 s, and the zero errors represent failed samples.

where K_p is the proportional gain, and τ is a time delay term, which enables larger gain values compared to pure P-control. This reduces static errors when positioning to a specific target point.

5. Measurements and results

5.1. Orientation adjustment

The first step in each measurement was to perform the orientation adjustment as in Subsection 3.1. An example

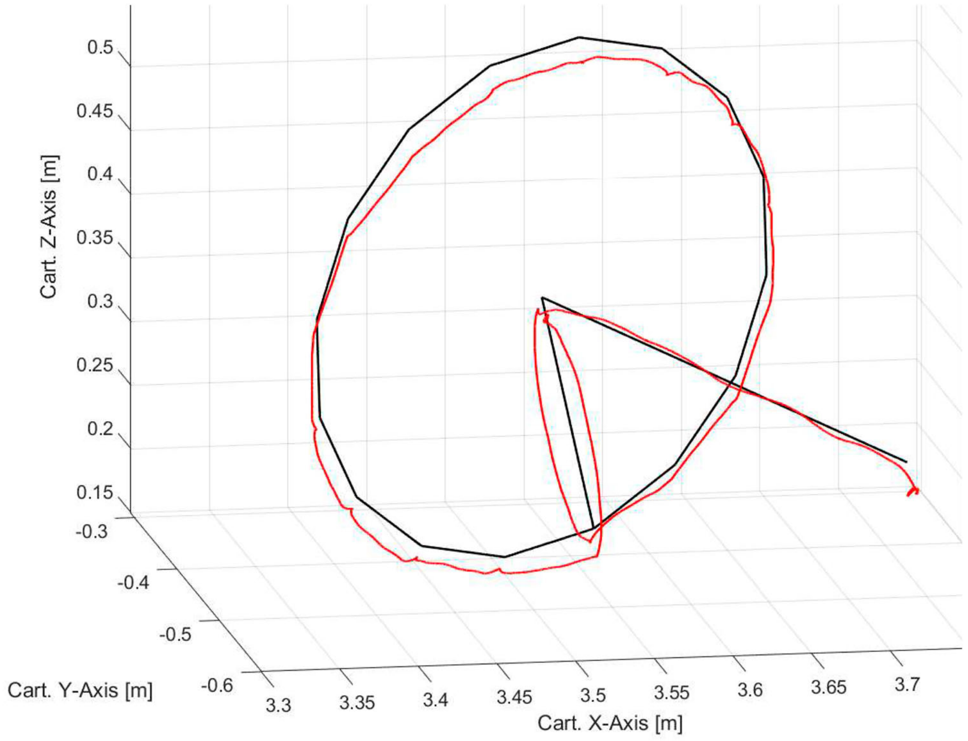

Figure 6. An example of the circular path and the computed line equation–based range adjustment. The black line shows the reference path, whereas the red line shows the completed path.

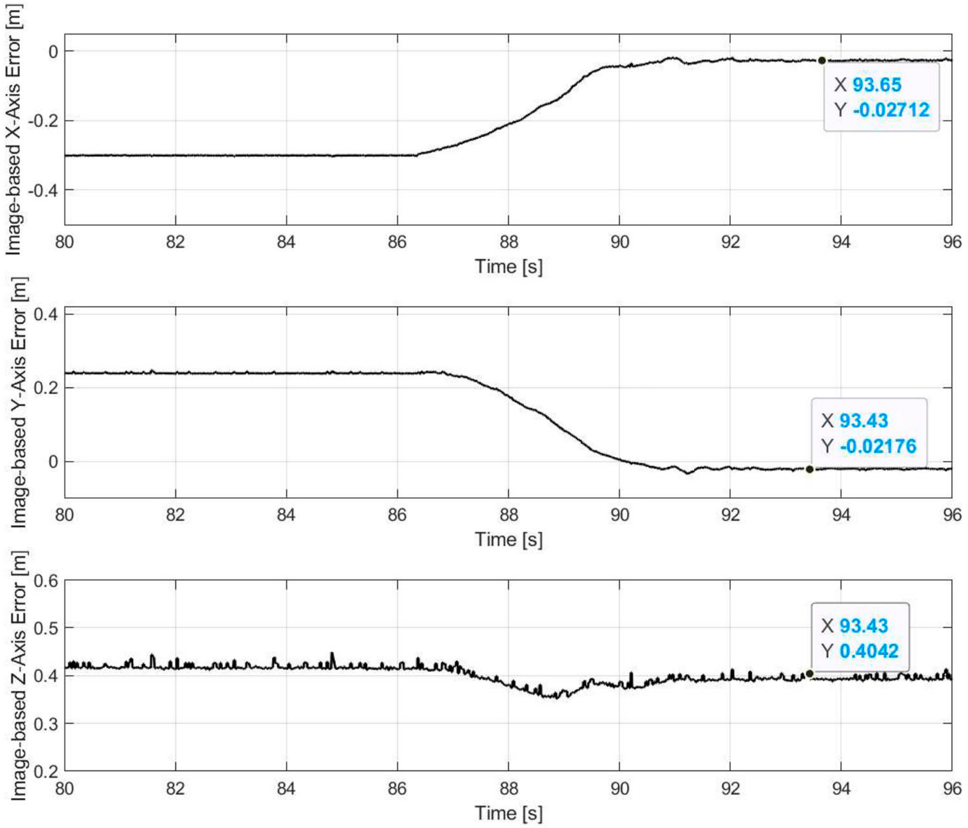

Figure 7. Image-based position errors between the tool marker and the OOI. The objective was to reach zero errors w.r.t. the *X*-axis and the *Y*-axis, but the depth along the *Z*-axis is also shown.

result is illustrated in Figure 5, which shows the image-based rotation errors between the tool marker and the OOI. The errors are expressed as XYZ Euler angles for clarity. As shown, two of the errors are successfully driven close to zeros by using a single control input. The objective was to achieve zero orientation errors so that the tool's torsion axis is perpendicular w.r.t. the OOI. The perpendicularity is important in the peg-in-hole task and for line equation–based range adjustment.

The measurements were performed repeatedly, with the error behaviors resembling the illustrated example. In general, the orientation signals were not of good quality, and the initial rotation difference was assumed to be small. The orientation adjustment also assumed the uniformity of the tool marker's frame and the TCP frame based on the kinematic model. In the experimental setup, some errors existed in the alignment.

5.2. Range adjustment

5.2.1. Line equation–based method

After the orientation adjustment, the line equation–based range adjustment was initiated by executing the circular path. The circular path in Equations (10)–(12) was realized using 18 points, with the first and last points set at the center of the circle. The radius was set to 20 cm. After the path was successfully completed, the reference TCP pose was obtained using Equation (16). This method was designed as a single point-to-point problem, as it is expected that the OOI will be occluded toward the end of the final approach.

An example execution of the circular path and driving to the reference TCP pose is illustrated in Figure 6. The black line represents the reference path, whereas the red line shows the completed path. Generally, this was the typical result achieved with the experimental setup. As shown, the completed path clearly resembles a circle; however, it is not completely planar as required to compute the line equation–based range adjustment as precisely as possible.

The image-based position errors between the tool marker and the OOI are illustrated in Figure 7. Notably, this method did not account for the depth parameter (Z-axis), yet it is still shown. The X-axis and Y-axis positioning errors, estimated by the camera, were reduced to a range of a few centimeters. The measurements were performed repeatedly, and the results are documented in Table 2, which also shows the image-based absolute rotation errors resulting from the previous orientation adjustment. The mean absolute rotation errors between the tool marker frame and the OOI frame were reduced to less than 2°, except for the Z-axis, which contained more errors. The mean absolute positioning accuracy w.r.t. the XY plane was less than 3 cm per axis, which did not satisfy the desired sub-centimeter accuracy. In measurement 7, the desired accuracy of the sub-centimeter range was achieved; however, the result was clearly not reliably reproducible. Based on the results, no clear connection between a successful orientation adjustment and the resulting final positioning accuracy can be seen. However, the performance also relies on accurate execution of the circular path, for example.

Table 2. The results for the line equation–based range adjustment.

	e_{θ_X} [deg]	e_{θ_Y} [deg]	e_{θ_Z} [deg]	e_X [mm]	e_Y [mm]	e_Z [mm]
meas. 1	1.40	2.75	14.25	27	21	404
meas. 2	0.71	0.94	3.11	37	66	334
meas. 3	0.72	1.33	5.91	45	42	321
meas. 4	2.40	1.52	8.59	4	28	387
meas. 5	2.03	2.21	11.59	4	11	342
meas. 6	2.83	1.50	9.18	12	23	611
meas. 7	0.77	1.83	1.82	3	2	338
meas. 8	1.28	1.25	3.50	27	29	303
meas. 10	1.74	1.08	2.07	36	37	424
mean	1.58	1.57	6.64	24	29	338

Notes: The first three columns show the image-based absolute rotation errors between the tool marker and the OOI after the orientation adjustment. The respective absolute position errors after the range adjustment are shown in the next two columns. The depth (along the Z-axis) between the tool marker and the OOI in each measured case is also documented in the last column.

A servo-type implementation variant was also tested by applying the currently measured variables to Equation (16) and then providing a second control input to the system. However, this did not improve the end result as the dictating factor is the computed directional unit vector.

5.2.2. Trajectory matching–based method

For the trajectory matching–based range adjustment, the manipulator was first aligned with the OOI using the orientation adjustment. However, the trajectory matching–based method itself is not reliant on the orientation adjustment, unlike the line equation–based method.

The motion-based local calibration was conducted by first completing the planar path designed using Equations (17)–(19). The two pose trajectories were obtained using the VO/SLAM algorithm and the kinematic model-based TCP computation with joint encoders. The resulting point sets were matched, and the reference pose for the TCP was obtained using Equation (22). An example of the completed path is shown in Figure 8. The respective mean and maximum absolute errors after the point set matching in a single measurement are documented in Table 3. As shown, the matching errors are small, and the calibrated camera signals correspond to the encoder-based TCP pose. The results were similar for the measured cases.

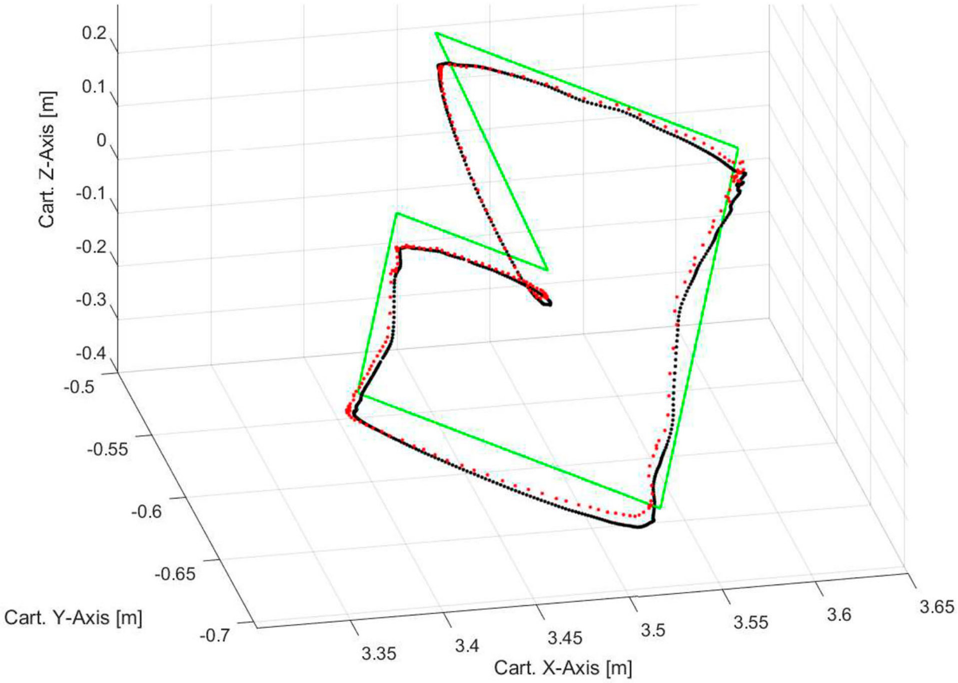

Figure 8. An example of trajectory matching. The green line denotes the reference path, the black line shows the encoder-based TCP position, and the red line illustrates the VO/SLAM-based camera position after the camera-to-kinematic model local calibration.

Table 3. The mean and maximum absolute errors after point set matching.

	Cartesian *X*-axis [mm]	Cartesian *Y*-axis [mm]	Cartesian *Z*-axis [mm]
mean error	3.7	1.8	14.2
maximum error	21.9	25.4	58.2

Notes: The errors are computed between the calibrated VO/SLAM point set and the encoder-based TCP point set.

After obtaining the local calibration, the manipulator was driven to the computed reference pose. An example result is shown in Figure 9, which shows the image-based position errors between the tool marker and the OOI. The final positioning is also shown in Figure 10, where the tool marker was driven to the OOI. As with the line equation–based method, the goal was to achieve precise positioning with a single point-to-point control input, meaning the camera would look and then move only once. However, two control inputs were required for precise positioning: The first was given at approximately 8 s, which did not result in the desired sub-centimeter accuracy. The second control input was given at approximately 45 s, which reduced the errors to the desired range, with the exception of the depth variable. It had an offset of $b = 10$ cm in Equation (21), and the depth is allowed more tolerance than the other axes, as positioning in the XY plane holds the most significance.

The measurements were performed repeatedly, starting from the orientation adjustment followed by the point set matching, and each measurement demonstrated uniform behavior, as shown in Table 4. Two updates for the visual control input were required in each measured case to achieve sub-centimeter position errors, meaning the camera was used to look and then move twice. For each case, the second update clearly resulted in sub-centimeter position errors, whereas with just one control update, the desired accuracy was not achieved, especially in the Y-axis direction. This suggests that the image-based pose estimation of the markers becomes more accurate as their distance to the camera is reduced, although the camera is relatively close initially.

Overall, the trajectory matching–based method performed much more reliably than the line equation–based method. The results demonstrate that sub-centimeter positioning accuracy can be achieved using a low-cost visual sensor, despite using the simple rigid-body kinematic modeling and controller structures that are prevalent in the industry. However, the trajectory matching–based method relies on the performance of the VO/SLAM pose estimation, which requires certain conditions, such as sufficient lighting and textured surfaces, for feature extraction. There exists some research on improving the performance of VO/SLAM in challenging environments, for example, [34].

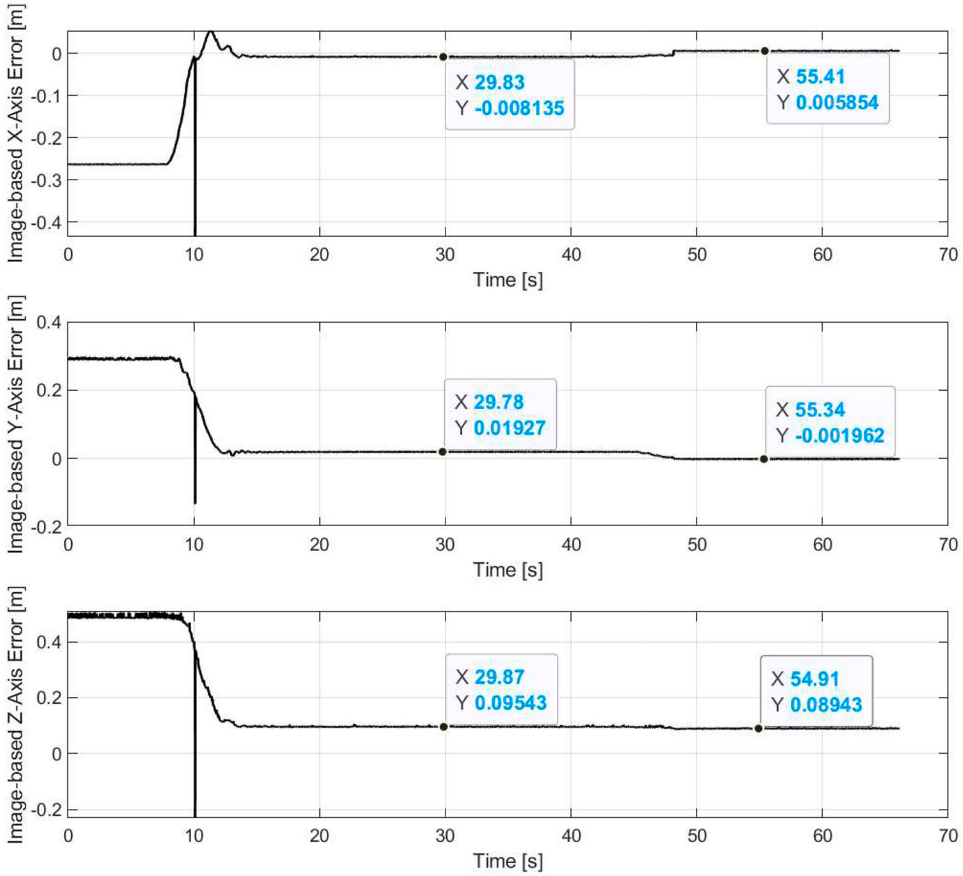

Figure 9. An example of image-based position errors before and after the trajectory matching–based range adjustment. Two control updates (manually input at approximately 8 s and 45 s) were required to achieve sub-centimeter position errors in the XY plane, with the depth axis error slightly larger.

Table 4. Image-based absolute positioning errors between the tool marker and the OOI using the trajectory matching–based method.

1st input	e_X [mm]	e_Y [mm]	e_Z [mm]	2nd input	e_X [mm]	e_Y [mm]	e_Z [mm]
meas. 1	8.1	19.3	95.4		5.9	1.9	89.4
meas. 2	11.4	30.4	67.3		2.9	3.0	112.7
meas. 3	6.4	29.4	110.1		5.1	0.8	83.8
meas. 4	4.1	35.2	91.8		5.7	1.1	105.9
meas. 5	0.4	27.6	91.5		3.6	0.7	93.9
meas. 6	5.9	26.2	111.0		2.8	0.9	79.8
meas. 7	11.3	22.3	91.8		0.5	0.2	79.0
meas. 8	6.0	25.6	110.9		1.0	0.8	97.6
meas. 9	17.1	30.8	103.9		0.6	0.8	97.6
meas. 10	5.6	26.8	85.5		4.9	1.5	89.2
mean	7.6	27.4	95.9		3.3	1.2	92.3

Notes: Two updates for the visual control input were required in each measured case to achieve sub-centimeter position errors, meaning the camera was used to look and then move twice. The first three columns show the errors after the first update, and the last three columns show the respective errors after the second update.

6. Conclusion

In this paper, a vision-based guidance methodology for precise TCP positioning of HDLR manipulators was examined. For HDLR manipulators, the conventional extrinsic camera calibration procedure is not realistic or practical due to the dynamic, unstructured working environments and inaccuracies in rigid-body kinematic modeling. Instead, local calibration and visual guidance based on direct measurements in the camera frame were proposed. Notably, placing predefined calibration objects in the environment is not required. The presented methodology comprised orientation adjustment followed by range adjustment, for which two methods were explored. A key assumption is that the tool and

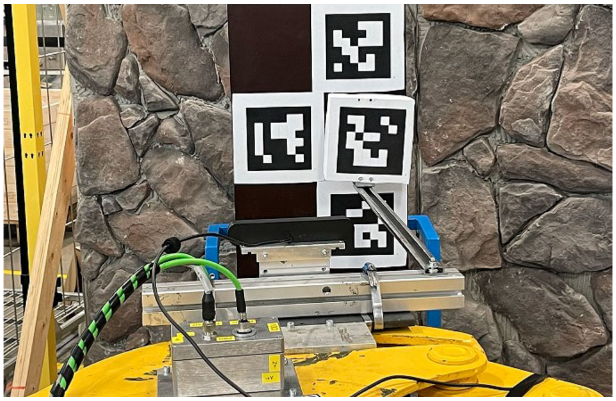

Figure 10. An example of the final positioning result with the trajectory matching–based range adjustment.

the OOI are detectable using a camera in the eye-in-hand configuration, and that their 6 DOF poses can be estimated. ArUco markers were used as 'generic' representations of a tool and an OOI. However, for HDLR manipulators in dynamic environments, it is not a realistic option to place markers around the workspace. Thus, for a practical application, the vision-based detection and pose estimation of tools and OOIs have to be realized with application specific parameters. This includes omitting the markers completely and shifting to real-world objects. This can be a challenging task depending on the complexity of the target objects, such as tools, tool racks, and drill holes.

Real-time experiments were conducted using a HDLR manipulator with a 5 m reach, and a low-cost stereo camera was used for vision-based measurements. Open-loop visual control was employed by first looking and then moving. The motivation was to avoid closed-loop visual control due to safety concerns. Nevertheless, sub-centimeter positioning accuracy was achieved using the trajectory matching–based method. For the application of interest, the desired positioning accuracy is +/− 5 mm, as the tolerances are much larger than in industrial robotics. The line equation–based method suffered from accumulated errors beginning from the hardware installation and orientation adjustment to the path tracking. Thus, the positioning accuracy with this method had considerable variation, and the desired accuracy was not reliably achieved in the experiments.

The main advantage of the proposed methods is that computing the pose error between a detected tool and an OOI directly in the image frame, using a low-cost camera, enables precise positioning accuracy. This is typically very challenging to achieve with HDLR manipulators due to their characteristic uncertainties. Moreover, the motion-based local calibration does not require calibration objects placed in the workspace. The main challenges lie in the robustness and reliability of vision-based algorithms. For example, VO/SLAM systems require certain conditions and the object detection and pose estimation for complex objects can be challenging to realize. Future research should focus on extending the proposed locally calibrated vision-based guidance method into a practical application by replacing the ArUco markers with application specific objects.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by the Technology Industries of Finland Centennial Foundation and the Jane and Aatos Erkko Foundation in Future Makers program.

Notes on contributors

Petri Mäkinen received his M.Sc. in Automation Engineering from Tampere University of Technology, Finland, in 2016 and his D.Sc. (Tech.) in Automation Technology and Mechanical Engineering from Tampere University, Finland, in 2023. His current research interests include vision-based control and machine learning in field-robotic applications.

Pauli Mustalahti received the M.Sc. degree in engineering from Tampere University of Technology in 2016 and the D.Sc. (Tech.) in Automation Technology and Mechanical Engineering from Tampere University, Finland, in 2023. He is a researcher in the unit of Automation Technology and Mechanical Engineering, Tampere University, Tampere, Finland. His research interests include nonlinear model-based control of robotic manipulators.

Sirpa Launis received her M.Sc. degree in mechanical engineering from Tampere University of Technology in 1999. She has over 20 years of experience in various stages of R&D with non-road mobile machinery, and is currently working as a product manager, tunneling, at Sandvik.

Jouni Mattila received his M.Sc. in 1995 and Dr. Tech in 2000, both from Tampere University of Technology (TUT), Tampere, Finland. He is currently a professor of machine automation with the unit of Automation Technology and Mechanical Engineering, Tampere University. His research interests include machine automation, nonlinear model-based control of robotic manipulators and energy-efficient control of heavy-duty mobile manipulators.

References

[1] Slamani M, Nubiola A, Bonev I. Assessment of the positioning performance of an industrial robot. Ind Robot Int J. 2012;39(1):57–68. doi: 10.1108/01439911211192501

[2] Cebollada S, Payá L, Flores M, et al. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Syst Appl. 2021;167:114195. doi: 10.1016/j.eswa.2020.114195

[3] Hutchinson S, Hager GD, Corke PI. A tutorial on visual servo control. IEEE Trans Rob Autom. 1996;12(5):651–670. doi: 10.1109/70.538972

[4] Singh A, Kalaichelvi V, Karthikeyan R. A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks. Cogent Eng. 2022;9(1):2050020. doi: 10.1080/23311916.2022.2050020

[5] Wilson WJ, Hulls CW, Bell GS. Relative end-effector control using Cartesian position based visual servoing. IEEE Trans Rob Autom. 1996;12(5):684–696. doi: 10.1109/70.538974

[6] Chaumette F. Potential problems of stability and convergence in image-based and position-based visual servoing. In: The confluence of vision and control. London: Springer; 2007. p. 66–78.

[7] Hafez AA, Cervera E, Jawahar C. Hybrid visual servoing by boosting IBVS and PBVS. In: 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications. IEEE; 2008. p. 1–6.

[8] Pan W, Lyu M, Hwang KS, et al. A neuro-fuzzy visual servoing controller for an articulated manipulator. IEEE Access. 2018;6:3346–3357. doi: 10.1109/ACCESS.2017.2787738

[9] Franaszek M, Cheok GS. Using locally adjustable hand-eye calibrations to reduce robot localization error. SN Appl Sci. 2020;2:1–13. doi: 10.1007/s42452-020-2626-2

[10] Flandin G, Chaumette F, Marchand E. Eye-in-hand/eye-to-hand cooperation for visual servoing. In: Proc. 2000 ICRA. Millennium Conf. IEEE Int. Conf. Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). Vol. 3. IEEE; 2000. p. 2741–2746.

[11] Wang CC. Extrinsic calibration of a vision sensor mounted on a robot. IEEE Trans Rob Autom. 1992;8(2):161–175. doi: 10.1109/70.134271

[12] Enebuse I, Foo M, Ibrahim BSKK, et al. A comparative review of hand-eye calibration techniques for vision guided robots. IEEE Access. 2021;9:113143–113155. doi: 10.1109/ACCESS.2021.3104514

[13] Jiang J, Luo X, Luo Q, et al. An overview of hand-eye calibration. Int J Adv Manuf Technol. 2022;119(1–2):77–97. doi: 10.1007/s00170-021-08233-6

[14] Pedrosa E, Oliveira M, Lau N, et al. A general approach to hand–eye calibration through the optimization of atomic transformations. IEEE Trans Robot. 2021;37(5):1619–1633. doi: 10.1109/TRO.2021.3062306

[15] Dornaika F, Horaud R. Simultaneous robot-world and hand-eye calibration. IEEE Trans Rob Autom. 1998;14(4):617–622. doi: 10.1109/70.704233

[16] Corke P, Roberts J, Winstanley G. Vision-based control for mining automation. IEEE Robot Autom Mag. 1998;5(4):44–49. doi: 10.1109/100.740465

[17] Kiziroglou ME, Boyle DE, Yeatman EM, et al. Opportunities for sensing systems in mining. IEEE Trans Ind Inform. 2016;13(1):278–286. doi: 10.1109/TII.2016.2636131

[18] Bonchis A, Duff E, Roberts J, et al. Robotic explosive charging in mining and construction applications. IEEE Trans Autom Sci Eng. 2013;11(1):245–250. doi: 10.1109/TASE.2013.2241425

[19] Xu J, Hou Z, Liu Z, et al. Compare contact model-based control and contact model-free learning: a survey of robotic peg-in-hole assembly strategies. arXiv preprint arXiv:190405240. 2019.

[20] Shen Y, Jia Q, Wang R, et al. Learning-based visual servoing for high-precision peg-in-hole assembly. In: Actuators. Vol. 12. MDPI; 2023. p. 144.

[21] Kivelä T, Mustalahti P, Mattila J. Real-time distance query and collision avoidance for point clouds with heavy-duty redundant manipulator. In: 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE; 2017. p. 272–277.

[22] Gallwey J, Eyre M, Coggan J. A machine learning approach for the detection of supporting rock bolts from laser scan data in an underground mine. Tunn Undergr Space Technol. 2021;107:103656. doi: 10.1016/j.tust.2020.103656

[23] Singh SK, Banerjee BP, Raval S. A review of laser scanning for geological and geotechnical applications in underground mining. Int J Min Sci Technol. 2022;33(2):133–154.

[24] Lampinen S, Niu L, Hulttinen L, et al. Autonomous robotic rock breaking using a real-time 3D visual perception system. J Field Robot. 2021;38(7):980–1006. doi: 10.1002/rob.v38.7

[25] Zhou S, Shen C, Pang F, et al. Position-based visual servoing control for multi-joint hydraulic manipulator. J Intell Robot Syst. 2022;105(2):33. doi: 10.1007/s10846-022-01628-x

[26] Lauer APR, Lerke O, Blagojevic B, et al. Tool center point control of a large-scale manipulator using absolute position feedback. Control Eng Pract. 2023;131:105388. doi: 10.1016/j.conengprac.2022.105388

[27] Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, et al. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. 2014;47(6):2280–2292. doi: 10.1016/j.patcog.2014.01.005

[28] Mäkinen P, Aref MM, Mattila J, et al. Application of simultaneous localization and mapping for large-scale manipulators in unknown environments. In: 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE; 2019. p. 559–564.

[29] Sciavicco L, Siciliano B. Modelling and control of robot manipulators. London: Springer-Verlag; 2000.

[30] Mäkinen P, Mustalahti P, Launis S, et al. Probabilistic camera-to-kinematic model calibration for long-reach robotic manipulators in unknown environments. In: 2022 IEEE 17th International Conference on Advanced Motion Control (AMC). IEEE; 2022. p. 48–55.

[31] Min Z, Wang J, Meng MQH. Robust generalized point cloud registration using hybrid mixture model. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2018. p. 4812–4818.

[32] Mur-Artal R, Tardós JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Trans Robot. 2017;33(5):1255–1262. doi: 10.1109/TRO.2017.2705103

[33] Jazar RN. Theory of applied robotics–kinematics, dynamics, and control. Dordrecht: Springer; 2010.

[34] Etxeberria-Garcia M, Zamalloa M, Arana-Arexolaleiba N, et al. Visual odometry in challenging environments: an urban underground railway scenario case. IEEE Access. 2022;10:69200–69215. doi: 10.1109/ACCESS.2022.3187209