Eye-In-Hand Pose Estimation of Industrial Robots

Christoph Buchner, Peter Gsellmann,
Martin Melik Merkumians, *Member, IEEE*, Georg Schitter, *Senior Member, IEEE*
Automation and Control Institute
TU Wien
Vienna, Austria
buchner@acin.tuwien.ac.at

***Abstract*—This paper proposes a pose estimation approach of an industrial six-degree-of-freedom robot without the need of externally placed sensors. An RGB-D camera that is placed at the robots end-effector is combined with a SLAM algorithm to act as a sensor system. The presented system based on the novel integration approach is evaluated by performing test trajectories and is capable of estimating the tool-center point with a standard deviation of 2.6 mm and performing a joint state estimation with a standard deviation of 13.6 mrad without any external sensor.**

***Index Terms*—Industrial Robots, SLAM, Visual Tracking, Performance Evaluations and Benchmarking**

I. Introduction

The tasks of modern robots are ranging from exploring hazardous environments such as the deep sea or the surface of Mars, supporting chores, up to the precise manufacturing of high-quality and cost-efficient products [1], [2]. Especially manufacturing relies heavily on the deployment of robots as they satisfy the desire for cost-effective production by avoiding manual labor costs, while facilitating precise assembly and machining operations [3].

In recent years, a lot of research activity was dedicated to ease the deployment of robots and increasing their feasibility for medium and small enterprises. In particular, the complex setup routines and the expensive hardware that are necessary to let multiple robots cooperate in production lines during high precision operations decreases the viability of robot-based manufacturing for small-scale businesses.

To overcome these deficiencies, cameras are often used to provide rich information of the current environment of the robot and are therefore utilized in various algorithms [4]–[7]. Additionally, since the number of affordable and specialized cameras for robotic systems has increased in recent years, they have become common components in newly developed systems [8].

There are two main configurations to place a camera in a robotic system [5]. In the eye-to-hand configuration, it is possible to directly estimate the current robot pose, as long as the field of view is not occluded. Using the appearance of the robot, a CAD model can be combined with an iterative closest point (ICP) optimization strategy to estimate the pose of robot links in the scenery [9].

However, the direct line of sight constraint between robot end-effector and camera leads to problems during commissioning of the system, as at installation sites an occlusion-free environment can not be guaranteed. Additionally expert knowledge for positioning the camera is necessary and calibration is a time consuming and expensive process.

In contrast to the eye-to-hand configuration, the state-of-the-art eye-in-hand pose estimation algorithms do not pursue to obtain the pose of the robot in the world reference frame, instead focusing on the camera-to-work-object relation [10], [11]. This approach has proven to give superior accuracy during object manipulation, as the accuracy of the sensor is utilized more efficiently. However, such systems are focused on a local description and the desired task may fail if the work object is obscured or not in the current field of view.

Other solutions of the pose estimation problem are present in the context of mobile robotics, where the current localization and navigation in an unknown environment is of interest. There, the robot is equipped with a camera observing the passing scenery and is inferring the pose transitions between consecutive images. Such systems are thereby able to deduct the current robot pose without any additional setup at installation site. Algorithms solving the aforementioned problem are referred to as simultaneous localization and mapping (SLAM) algorithms [12]–[15]. However, SLAM-based systems have mainly been used to estimate the pose of mobile robots in large exploration scenarios and only limited research is conducted in heavily limited environments, as it is the case for most industrial robots.

By directly deploying a SLAM algorithm in the joint space of the robot, the direct estimation of the joint configuration is possible [16], however these approaches only estimate the camera motion that is described by the kinematic model and thus fail to predict mechanical imperfections such as link deflections. Therefor, a system that performs the estimation in the task space of the robot is desired.

The contribution of this paper is a joint and 6-DoF pose estimation of an industrial robot without external sensors and the evaluation of the performance achieved by the proposed system.

The remainder of this paper is organized as follows: Section II gives an overview of the components and algorithms deployed to successfully estimate the robot pose solely via visual sensing. In Section III, the additional components necessary to evaluate the performance are described. Subsequently, Section IV presents the results of the experiments, followed by an analysis in Section V. Finally, the findings of this paper are summarized and an outlook is given in Section VI.

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053

II. System Overview

The system is designed to estimate the robot's pose, given the visual clues obtained from a moving camera, assuming that the camera to end-effector relation is known. In contrast to the eye-to-hand estimation approach, the camera is placed at the robot's end-effector, forming an eye-in-hand configuration. The images taken by the camera are passed to two separate systems:

- the SLAM system which performs the relative pose estimation of the camera motion, and
- a fiducial marker detection solely for the purpose of initializing the camera poses.

To evaluate the system performance the information on the current camera pose is forwarded to an inverse kinematic solver, which calculates the current joint configuration of the robot. The overall system and its components are illustrated in Fig. 1 and presented in the following sections.

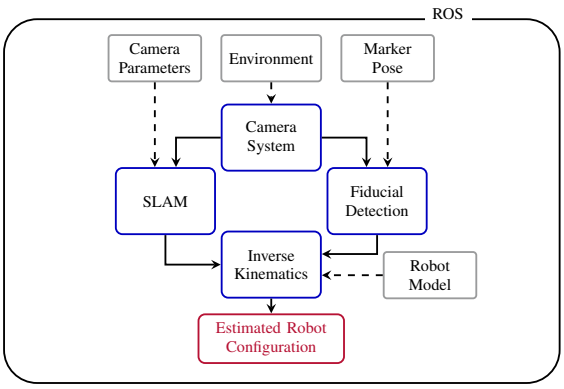


Fig. 1. Data flow and system composition of the proposed SLAM-based eye-in-hand pose estimation approach. Each blue rectangle represents an independent task implemented and executed in the Robot Operating System (ROS) environment. The gray boxes are the input data of the system and the output is represented in red. Black lines between tasks represent the communication using ROS messages. Dashed lines show the relationship between tasks and their input data.

A. SLAM

The SLAM algorithm is responsible for inferring the camera motion based on the passing scenery observed by the camera. A SLAM algorithm usable in the proposed setup needs to estimate the camera motion online, i.e. the algorithm must be capable of extracting the movement information from subsequent images and publish it at a guaranteed rate. It is also desirable that the algorithm is usable in various system configurations with different kinds of cameras deployed. In addition to the constraints above, the presented use case does not allow for manipulations of the environment, thus SLAM algorithms using natural environment features are preferred over those that use artificial features.

Therefore, the state-of-the-art ORB SLAM 3 [13] is used. It guarantees a worst-case update frequency of 30 Hz and is able to handle stereo, RGB-D, and monocular cameras. Pinhole and fish-eye distorted camera models are also supported, allowing for even more diversity in its application.

B. Fiducial Detection

As the SLAM algorithm only provides information about the relative pose of the camera, a second system to initially anchor the pose of the camera is necessary. In this paper, the anchor is realized by placing a marker on robots base plate. During the manufacturing and assembly process of a robot, it is possible to place fiducial markers with high accuracy. Therefore, the system proposed in this paper assumes that a fiducial marker is placed on the robot's base plate with a known relation to the robot's first link, as can be seen in Fig. 2a. However, this fiducial marker can be realized as the robot manufacturer logo or any other recognizable image. For simplicity, a ChArUco [17] board is used in combination with the OpenCV [18] library's implementation of a ChArUco detection algorithm.

C. Inverse Kinematics

Subsequent to the initialization of the camera pose using the fiducial detection, the static transformation between the end-effector and the camera frame is used to express the current end-effector pose. After the end-effector pose is calculated, an inverse kinematic algorithm is deployed to solve for the current joint state $\boldsymbol{q}_C$. The Levenberg–Marquardt algorithm is utilized to calculate the update of the current joint state

$$\boldsymbol{\Delta}_q = \left[J(\boldsymbol{q}_C)^T J(\boldsymbol{q}_C) + \lambda I\right]^{-1} J(\boldsymbol{q}_C)^T \boldsymbol{\Delta}_e(\boldsymbol{q}_C), \quad (1)$$

using the Jacobian matrix $J(\boldsymbol{q}_C)$ of the error vector $\boldsymbol{\Delta}_e(\boldsymbol{q}_C)$ and the damping factor λ. Thereby, the error vector

$$\boldsymbol{\Delta}_e(\boldsymbol{q}_C) = \mathtt{vec}(T_D^{-1}\, T_C(\boldsymbol{q}_C) - I) \quad (2)$$

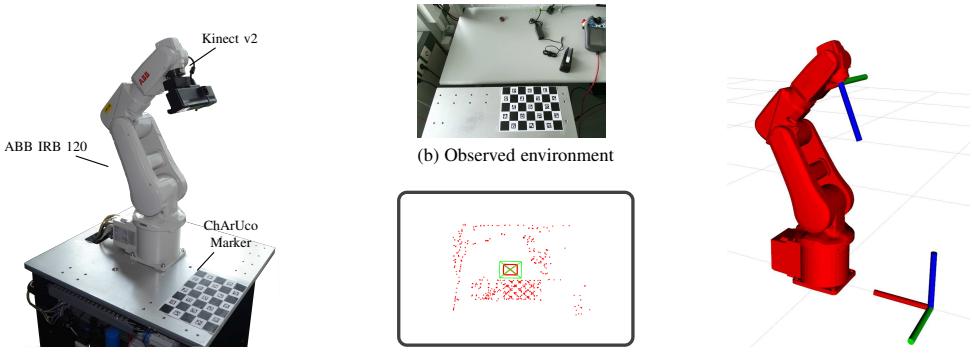
is calculated by using the $\mathtt{vec}()$ operator to reorder the matrix error terms originating between the homogeneous transformation of the estimated end-effector pose T_D and the current transformation of a simulated model $T_C(\boldsymbol{q}_C)$, which depends on the current joint state $\boldsymbol{q}_C$ and the forward kinematics of the deployed robot.

Combining the inverse kinematic scheme of Section II-C with the SLAM-based camera pose estimation, an eye-in-hand pose estimation of an industrial robot is achieved. To verify the system performance, experiments on an industrial robot are carried out with different pre-planned motion trajectories. Therefore, the hardware and additional components necessary to perform the evaluation are specified in more detail in the next section.

III. Experimental Verification of the Proposed Pose Estimation

The proposed system is verified using an ABB IRB 120 industrial robot equipped with a single Kinect v2 RGB-D camera at its end-effector. The software components of

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053



(a) Experimental setup (b) Observed environment (c) ORB SLAM 3 [13] map (d) Virtual RVIZ [19] model

Fig. 2. System overview: (a) Hardware setup for the experiments. The ChArUco marker serves as a substitute for an appearance based initialization of the camera pose, the industrial robot ABB IRB 120 is used as a stiff evaluation platform to execute a pre-defined motion and the Kinect v2 3D time-of-flight camera observes the scenery. (b) Observed environment, which is then forwarded to the ORB SLAM 3 algorithm. (c) Generated map from the ORB SLAM 3 algorithm. (d) Virtual robot model based on the estimated camera pose to visualize the current estimate of the robot pose.

Section II are executed on a PC equipped with an ASUS GeForce™ RTX 3070 GPU, an AMD Ryzen™ 9 3900X CPU, and a 32 GB DDR4 3200 MHz memory. The specific setup and the configuration of the robot and camera are specified in the following.

A. Robot System

An ABB IRB 120 is chosen as the robot platform to perform the experiments. This robot is designed as an anthropomorphic human arm robot, supporting end-effector motion in all six degrees of freedom (DoF), featuring a rigid structure and integrated angular encoders with an accuracy of up to 174 µrad. The ground-truth of the robot pose is then expressed by combining the measured joint angles with the Denavit-Hartenberg parameters [20]. However, to communicate with the ABB IRB 120 robot via ROS, an additional open source software module `open_abb` available in the GitHub repository [21] is utilized.

B. Camera System

The camera system indicated in Fig. 1 is the main sensor in the overall setup. As introduced in Section II, the utilized SLAM system limits the usable cameras. Using the ORB SLAM 3 software library, it is possible to support monocular, stereo, and RGB-D cameras. However, the current implementation is restricted to a single camera. Within this setup, a Kinect v2 [22] 3D time-of-flight camera is used in combination with the open-source `iai_kinect2` driver [23].

The camera was calibrated internally and externally prior to the experiment using the ChArUco calibration algorithm of the OpenCV library [24].

IV. System Evaluation

Four scenarios are considered to provide a systematic overview of the performance of the system. First, a translation ($\mathrm{Move}_{\mathrm{lin}}$) is executed in each of the directions of the end-effector frame with a range of ±0.1 m to evaluate the tracking performance in linear motion scenarios. Second, a path consisting of rotations around the robot base ($\mathrm{Move}_{\mathrm{rot}}$) frame in a range of ±314 mrad is carried out to evaluate the performance in the case of camera rotations. In these two scenarios, the crosstalk between the estimated translation and rotation is evaluated in addition to the interference between different rotations or translations. Third, a path of actuating each joint separately about ±0.5 rad ($\mathrm{Move}_{\mathrm{J}}$) is executed, which is coarsely equivalent to an joint space coverage rate of 60 %, to evaluate the interference between the estimated joints. Fourth, the joints are actuated randomly ($\mathrm{Move}_{\mathrm{rnd}}$) to evaluate the tracking performance of non-predefined motions with a complex composition of rotations and translations in the camera frame.

The following sections present in detail the results of the linear translation experiment in the operation space and configuration space. Subsequently, the root mean square (RMS) error of the proposed test cases are presented.

A. Pose Estimation Performance

In this experiment, the robot is initialized with its head-down configuration, which means that the camera is facing the robot base plate as in Fig. 2. The robot is actuated so that the camera performs a linear translation in the direction of the X-Axis with a speed of approximately $10\,\mathrm{mm\,s^{-1}}$. After 10 s, velocity is increased to $25\,\mathrm{mm\,s^{-1}}$, which can be seen in Fig. 3, since the slope of the X-Axis motion is steeper afterward. Simultaneously with the actuated translation, the end-effector orientation is monitored to enable the evaluation of the cross-coupling between translation and rotation. During the experiment a refresh rate of 30 Hz was accomplished based on the performance ORB SLAM3.

B. Joint Estimation Performance

The end-effector pose is expressed by transforming the camera pose to the end-effector frame. The inverse kinematic solver from Section II is then applied to regain information about the robot's joint state, allowing a direct comparison

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053

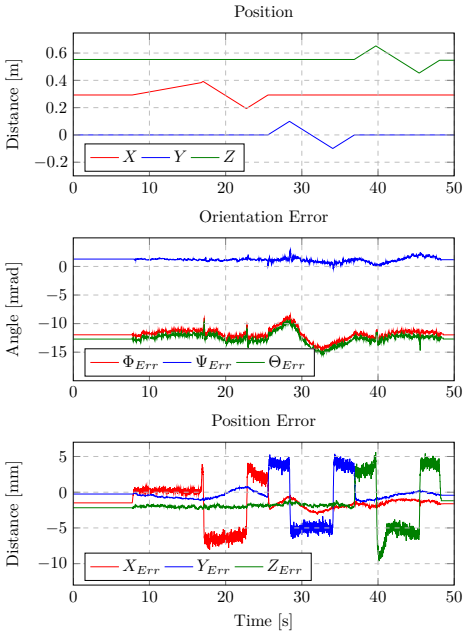


Fig. 3. Measured and estimated trajectory of the robot's end-effector pose during a linear translation in each direction. The position is represented by the euclidean coordinates for the X-Axis, Y-Axis and the Z-Axis. The orientation is expressed by the variables Φ, Ψ, and Θ, which represent the angle enclosed by the axes of the current end-effector frame and the end-effector frame of the initial orientation. Four main error sources are identified within this figure a bias error is observable at the start time, a position error during the motion, a cross talk between the active and passive axes and an interference between rotation and linear motion.

with the built-in angular encoders. It is shown in Fig. 4 that the estimated joint angles follow the ground-truth trajectory – even in the case of a nonlinear joint motion originating from a linear translation or a rotation of the end-effector.

In the next section, a further evaluation of the results shown in Figs. 3 and 4 is carried out by computing the overall RMS error during each experiment.

C. Error Evaluation

The results of the experiments are summarized in Table I. Each of the columns Move_{lin}, Move_{rot}, Move_{J} and Move_{rnd} represent one of the four evaluation scenarios. The evaluation is carried out by calculating the RMS error

$$e_{RMS} = \sqrt{\frac{1}{n_x}\sum_k (\hat{f}_k - f_k)^2}, \tag{3}$$

where n_x is the number of samples, f_k is the ground truth acquired from the angular encoders, $\hat{f}_k$ is the estimated value and the index k is the discrete sampling time. By combining the RMS error terms of all scenarios, the overall estimation performance is given by

$$\overline{\text{Moves}} = \sqrt{\frac{1}{n}\sum_x n_x \,\text{Move}_{\text{x}}{}^2}, \text{ with } \quad n = \sum_x n_x, \tag{4}$$

TABLE I
RMS ERROR OF THE ESTIMATED POSE EVALUATED BY PERFORMING LINEAR TRANSLATIONS, ROTATIONS AROUND THE BASE FRAME, SEPARATE ACTUATION AND RANDOM ACTUATION IN EACH JOINT

Joint Angle RMS Error [mrad]

	Move_{lin}	Move_{rot}	Move_{J}	Move_{rnd}	$\overline{\text{Moves}}$
Joint_1	7.27	16.06	10.10	14.45	12.46
Joint_2	13.78	13.67	30.90	10.17	18.94
Joint_3	21.34	16.09	56.07	31.52	34.83
Joint_4	3.41	26.60	13.23	16.70	17.13
Joint_5	20.11	14.98	24.84	43.17	27.88
Joint_6	7.00	30.03	13.59	20.58	19.74
$\overline{\text{Joints}}$	13.90	20.56	29.38	25.40	23.05

TCP RMS Error [mm]/[mrad]

	Move_{lin}	Move_{rot}	Move_{J}	Move_{rnd}	$\overline{\text{Moves}}$
X	2.60	3.70	6.53	14.31	8.18
Y	2.23	4.65	2.78	9.19	5.45
Z	2.90	4.26	7.72	14.99	8.81
$\overline{\text{TR}}$	2.59	4.22	6.05	13.09	7.62
Φ	11.88	10.79	8.11	14.16	11.13
Ψ	1.23	11.61	10.53	9.06	9.43
Θ	12.60	11.28	8.07	14.81	11.53

in which Move_{x} represents the RMS error in the different scenarios and n_x are the number of samples per scenario. In the upper half of Table I, the RMS error for each joint is given, followed by the average RMS error for all joint estimates. The lower half concerns the tool-center point (TCP) RMS error: first for the separate Cartesian axes and subsequently the average RMS error over all translations $\overline{\text{TR}}$.

1) Crosstalk between rotation and translation: In experiments consisting of pure rotation or translation, a performance estimation of the desired motion is performed by analyzing the crosstalk between these two types of motion. In the first experiment, in which only translations are performed, the maximal observed crosstalk is 34 mrad m^{-1} and in the second experiment, which consists of rotations, the crosstalk is around 14 mm rad^{-1}.

2) Interference in orientation and translation: Similar to the evaluation of crosstalk, a coupling to the same type of motion is observed. This type of error, also referenced as interference, has been observed with a maximum magnitude of 0.3 % in case of the translation shown in Fig. 3 and during the second experiment consisting of only rotations with a value of 1.5 %.

3) Interference in the configuration space: Using the results of experiment three, in which only a single joint is actuated at a time, the worst interference occurred between joints six and two with a magnitude of 12 %.

4) Bias Evaluation: To further classify the error that arises during estimation, a separation of the RMS error into mean error

$$e_\mu = \frac{1}{n_x}\sum_k (\hat{f}_k - f_k), \tag{5}$$

and standard deviation

$$e_\sigma = \sqrt{\frac{1}{n_x - 1}\sum_k \left((\hat{f}_k - f_k) - e_\mu\right)^2} \tag{6}$$

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053

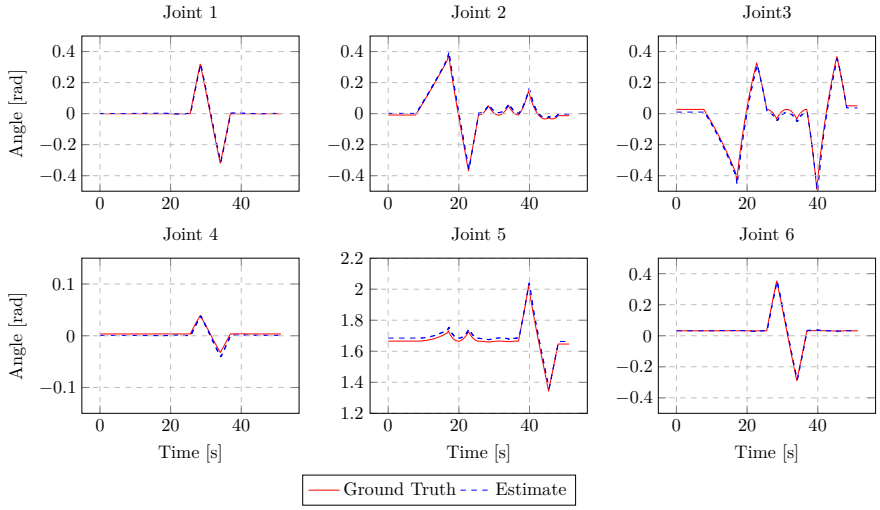


Fig. 4. The measured and estimated trajectory of a linear translation expressed in the robot configuration space. Joint angles are calculated using the current estimate of the end-effector pose followed by the calculation of the inverse kinematics. Although the nonlinear mapping of the inverse kinematic is applied the tracking of the ground truth data is visible.

is performed, where n_x is the number of samples, f_k is the ground truth acquired from the angular encoders, $\hat{f}_k$ is the estimated value, and k is the discretized sampling time. The remaining mean error in the TCP μ_{TCP} = 7 mm and the joints μ_{J} = 16 mrad represent the bias in the estimation process. The existence of the bias can be explained by erroneous calibration data, for example, in the manually measured marker pose, by a residual error in the camera calibration, or by an error in the robot kinematics. As this error is systematic, reappearing each time the same trajectory is executed, the SLAM-based approach is not capable of detecting and correcting these deviations.

V. Discussion

A. Absolute Performance

With the results presented in Table I and the experiment utilizing a random path, it is verified that an arbitrary motion of the end-effector is estimated as well as the designed paths that only consist of rotations and translations. Using the results from the experiments containing the trajectories involving pure rotations and translations, the crosstalk and interference between these two types of motion are observable. Comparing the estimates with the ground-truth data acquired by the angular decoders of the deployed industrial robot shows that the estimation of the current joint state leads to an average tracking RMS error of 23 mrad with a standard deviation of 13.64 mrad in each joint and a TCP estimation RMS error of 8 mm with a standard deviation of 2.6 mm on average.

B. Error Sources: SLAM

As the core component of this system, the SLAM algorithm has a major impact on the quality of pose estimation. Due to erroneous measurements or trajectory estimations, a wrong map point may be created, which impairs the actual and future camera poses. As a consequence, the estimated joints will be subject to errors, leading to a severe degeneration of the estimation performance, meaning that the RMS error increases. However, since SLAM algorithms are designed to cope with erroneous measurements, the system is able to recover from such faulty states if the already mapped scenes are revisited. The state correction that is subsequent to such a revisit is visible as a jump in the current pose estimation of the camera.

C. Error Sources: Calibration Data

The remaining bias, evaluated in Section IV-C4, can be explained by the remaining error in the calibration parameters. These parameters include the kinematic robot model of the deployed robot with its camera pose relative to the end-effector and the fiducial marker pose relative to the robot base. All of these must be provided beforehand and will negatively affect performance if they are not chosen correctly. In addition to bias, interference and crosstalk measured in Sections IV-C1 to IV-C3 can also be explained by the presence of a remaining calibration error. For instance, if there is a remaining camera barrel distortion, then a part of the translation is incorrectly estimated as rotation. Furthermore, any error in the calibration data will lead to systematic errors which cannot be mitigated by the current implementation, as they are not observable by the SLAM system.

D. Environmental Influence

In addition to the parameters mentioned above, environmental conditions are another important impact factor on the quality of the estimate. Due to the use of a vision-based sensor

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053

principle, there is a dependency on the current illumination and the uniqueness of the observed scenery. High illumination will improve pose estimation by allowing for a shorter exposure time of the camera sensor, thereby limiting the effect of motion blurs impairing the image.

In summary, the proposed system enables the estimation of the pose of an industrial six-degree-of-freedom robot. The experimental setup, consisting of a Kinect v2 RGB-D camera mounted on an ABB IRB 120 robot, achieves a standard deviation of the pose estimate of 2.6 mm, forming a basis for the prediction of mechanical imperfections in robotic systems and monitoring the system's condition.

VI. Conclusion

Images and depth information have been captured by a Kinect v2 time-of-flight camera mounted on the TCP of an ABB IRB 120 industrial robot, which are used to infer the motion of its end-effector using the ORB SLAM 3 algorithm to estimate consecutive poses.

To initially anchor these poses, a known ChArUco fiducial marker, placed on the base plate of the robot, is used as a substitute for an appearance-based initialization.

The knowledge of the current global camera pose is then used to deduce the current joint state by implementing an inverse kinematics algorithm based on a Levenberg-Marquardt nonlinear least-squares solver.

To further analyze the performance of the proposed system, four motion patterns are evaluated. In each of the experiments performed, the estimated end-effector poses and joint states are compared with the ground truth data acquired from the angular joint encoder of the robot. A performance evaluation is carried out, showing that a TCP standard deviation of 2.6 mm and a joint estimation standard deviation of 13.6 mrad is achievable.

The results presented in this work show that an eye-in-hand pose estimation algorithm is suitable to estimate the robot pose of a six degree-of-freedom robot by using only a single camera at its end-effector, and the kinematic chain model of the robot.

For future work, it is planned to utilize the proposed system in the context of modular robots to provide TCP measurements for algorithms such as collision avoidance or human-machine interaction. Furthermore, its feasibility as a complementary measurement system to satisfy safety requirements in sensitive environments is going to be evaluated.

References

[1] Y. Zhang, C. Bian, and J. Gao, "An Unscented Kalman Filter-based Visual Pose Estimation Method for Underwater Vehicles," in *International Conference on Unmanned Systems (ICUS)*, vol. 3rd, 2020, pp. 663–667.

[2] M. Bajracharya, M. W. Maimone, and D. Helmick, "Autonomy for Mars Rovers: Past, Present, and Future," *Computer*, vol. 41, no. 12, pp. 44–50, Dec. 2008.

[3] J. H. Jung and D.-G. Lim, "Industrial robots, employment growth, and labor cost: A simultaneous equation analysis," *Technological Forecasting and Social Change*, vol. 159, p. 120202, 2020.

[4] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-spatial information science*, vol. 21, no. 1, pp. 21–32, 2018, publisher: Taylor & Francis.

[5] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[6] L. S. Scimmi, M. Melchiorre, S. Mauro, and S. P. Pastorelli, "Implementing a Vision-Based Collision Avoidance Algorithm on a UR3 Robot," in *2019 23rd International Conference on Mechatronics Technology (ICMT)*, Oct. 2019, pp. 1–6.

[7] D. Burschka and G. Hager, "Vision-based control of mobile robots," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, May 2001, pp. 1707–1713 vol.2, iSSN: 1050-4729.

[8] S. Giancola, M. Valenti, and R. Sala, *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer, 2018.

[9] T. Varhegyi, M. Melik-Merkumians, M. Steinegger, G. Halmetschlager-Funek, and G. Schitter, "A Visual Servoing Approach for a Six Degrees-of-Freedom Industrial Robot by RGB-D Sensing," in *OAGM & ARW Joint Workshop Vision, Automation and Robotics*, 2017.

[10] J. Pyo, J. Cho, S. Kang, and K. Kim, "Precise pose estimation using landmark feature extraction and blob analysis for bin picking," in *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, vol. 14th, 2017, pp. 494–496.

[11] I. Ali, O. J. Suominen, E. R. Morales, and A. Gotchev, "Multi-View Camera Pose Estimation for Robotic Arm Manipulation," *IEEE Access*, vol. 8, pp. 174 305–174 316, 2020.

[12] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.

[13] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[14] M. Sewtz, X. Luo, J. Landgraf, T. Bodenmüller, and R. Triebel, "Robust Approaches for Localization on Multi-camera Systems in Dynamic Environments," in *International Conference on Automation, Robotics and Applications (ICARA)*, vol. 7th, 2021, pp. 211–215.

[15] R. Muñoz-Salinas and R. Medina-Carnicer, "UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," *Pattern Recognition*, vol. 101, p. 107193, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319304923

[16] M. Klingensmith, S. S. Sirinivasa, and M. Kaess, "Articulated Robot Motion for Simultaneous Localization and Mapping (ARM-SLAM)," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1156–1163, Jul. 2016.

[17] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[18] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *International Convention MIPRO*, vol. 35th, 2012, pp. 1725–1730.

[19] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, no. 2, pp. 337–345, 2015, publisher: Springer.

[20] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, 1955, publisher: ASME International.

[21] M. Dawson-Haggerty, "open-abb-driver," Oct. 2022, original-date: 2012-05-21T18:44:15Z. [Online]. Available: https://github.com/robotics/open_abb

[22] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 388–394.

[23] L. Xiang and Echtler, "OpenKinect/libfreenect2:," 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5167182

[24] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

Post-print version of the article: Christoph Buchner, Peter Gsellmann, Martin Melik Merkumians,and G. Schitter, "Eye-In-Hand Pose Estimation of Industrial Robots," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023. DOI: 10.1109/IECON51785.2023.10312053