Robotica (2024), **42**, pp. 3211–3229
doi:10.1017/S026357472400136X

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

High accuracy hybrid kinematic modeling for serial robotic manipulators

Marco Ojer[1,2], Ander Etxezarreta[1], Gorka Kortaberria[3], Brahim Ahmed[3], Jon Flores[3], Javier Hernandez[1], Elena Lazkano[2] and Xiao Lin[1]

[1]Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Donostia-San Sebastian, Spain
[2]University of the Basque Country (UPV/EHU), Donostia-San Sebastian, Spain
[3]Tekniker Research Centre, Basque Research and Technology Alliance (BRTA) Eibar, Eibar, Spain
Corresponding author: Marco Ojer; Email: mojer@vicomtech.org

Received: 16 November 2023; **Revised:** 16 May 2024; **Accepted:** 3 July 2024; **First published online:** 19 September 2024

Keywords: topological modeling of robots; robot dynamics; manufacturing; automation; serial manipulator design and kinematics

Abstract

In this study, we present a hybrid kinematic modeling approach for serial robotic manipulators, which offers improved accuracy compared to conventional methods. Our method integrates the geometric properties of the robot with ground truth data, resulting in enhanced modeling precision. The proposed forward kinematic model combines classical kinematic modeling techniques with neural networks trained on accurate ground truth data. This fusion enables us to minimize modeling errors effectively. In order to address the inverse kinematic problem, we utilize the forward hybrid model as feedback within a non-linear optimization process. Unlike previous works, our formulation incorporates the rotational component of the end effector, which is beneficial for applications involving orientation, such as inspection tasks. Furthermore, our inverse kinematic strategy can handle multiple possible solutions. Through our research, we demonstrate the effectiveness of the hybrid models as a high-accuracy kinematic modeling strategy, surpassing the performance of traditional physical models in terms of positioning accuracy.

1. Introduction

In the context of the fourth industrial revolution (Industry 4.0), the use of robot manipulators is being more and more extensive [1]. In the coming years, the robotics market is projected to grow and replace many industrial applications that usually have been performed by manual operations or traditional manufacturing machines. The use of robots is taking key importance in the classic strategic sectors such as aeronautics, automation, railway, etc. [2], as well as in new emerging sectors such as renewable energies [3] or health industries [4]. They are also being implemented in cutting-edge manufacturing processes such as additive manufacturing [5]. These highly repetitive manipulators provide great flexibility and adaptability and, thus, allow the automation of processes to a large extent.

However, many industrial tasks such as electronic board mounting, assemblies, precise measuring, or welding require accuracy levels not available in industrial manipulators. Moreover, repeatability, often referred to as precision, and accuracy, are two different concepts, yet they are commonly mistaken for one another, see Figure 1. Robotics manufacturers only ensure repeatability values. This metric is displayed in robot datasheets, as its value is highly dependent on the robot encoder's resolution and mechanical setup. On the other hand, accuracy is related to the mathematical models of the system mechanism and, in turn, relies on the approximated physical parameters of the robot. Most common approaches only take into account dimensional parameters such as Denavit-Hartemberg (DH) parameters or the modified DH convention [6, 7]. However, the position of the end effector is affected by many physical parameters such as plastic and elastic deformation, inertia momentum, gear backslash, joint compliance,

© The Author(s), 2024. Published by Cambridge University Press.

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

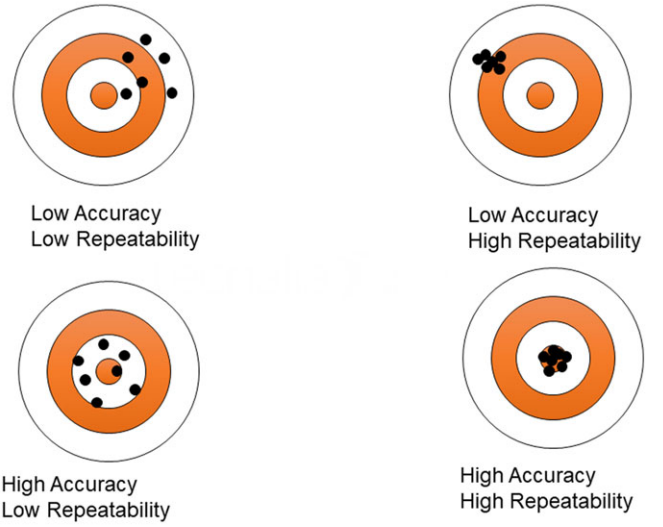


Figure 1. *Comparison of accuracy and repeatability/precision.*

link deflection, mechanical drifts due to temperature variation, etc. Thus, accuracy tolerance values are higher than repeatability. While modern industrial robots usually offer high levels of repeatability in spatial positioning, typically ranging around 20–50 μm, their accuracy is often limited [8], displaying values over than 1 mm in a $1m^3$ volume, which is insufficient for certain applications.

In typical industrial environments where target poses and trajectories are predefined, the prevalent methodology is to teach those target poses by manually guiding the robot to the desired pose and recording the joint values, thus ensuring repeatability tolerances or even positioning corrections. Nonetheless, this approach is not feasible for intelligent robotic applications where the target pose is unknown and variable, and it is deduced via sensors. In these cases, forward and inverse kinematics calculations are required to map the desired Cartesian pose to the corresponding joint values. Regardless, traditional kinematics relies on mathematical models, which inherently introduce modeling errors, leading to accuracy degradation.

Our main contribution is the development of a hybrid kinematic modeling scheme, aimed at enhancing the accuracy of robots to support intelligent robotics applications. This novel approach combines the geometric model of the physical system with a regression artificial neural network trained on accurate ground truth data. The underlying concept is that the neural network can accurately model deviations that cannot be analytically modeled. These models are then evaluated and validated using ground truth data, encompassing both virtual and real-world scenarios. The primary motivation behind our research is to advance the accuracy of robots, thereby enabling them to effectively perform a broader range of tasks.

To the best of our knowledge, this work pionEers the utilization of a hybrid kinematic formulation that includes the rotational component of the end effector. Furthermore, our research provides forward and inverse kinematics solvers, where the inverse kinematics is able to cope with the multiple solutions.

The paper is organized as follows. Section 2 reviews the related work. In Section 3, the methodology is described in detail. Sections 4 and 5 explain the experiments performed to validate our approach for forward and inverse kinematics, respectively. In section 5, the discussion is presented. Finally, section 6 outlines conclusions and future work.

2. Related work

There exist multiple calibration methods to improve the positioning accuracy of a robot. A first categorization can be made depending on whether they are parametric or not. Although parametric approaches

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

depend on a kinematic model which has to be characterized and adjusted [9–12], non-parametric approaches compare the robot positioning against a ground truth technology and estimate the corrections for the robot pose. The parametric method offers a notable advantage. Once the kinematic parameters that enhance the robot's accuracy are determined, its control implementation becomes straightforward. As a result, the correction in the Tool Center Pose (TCP) extends volumetrically across the robot's entire workspace, leading to accuracies up to 10 times greater than those achieved without calibration. However, parametric strategies require detailed model of the robot kinematics and its behavior as a mechatronic system. Alternatively, non-parametric methods require constant feedback or multiple testing so that the response is not immediate or requires a reconfiguration every time a new task is created. In this case, a map of corrections is obtained, where the points and orientations of the TCP of the robot are corrected with accuracies in the order of ten microns.

A second categorization dealing with robot positioning calibration is whether this is carried out before in-process operation (offline) or done directly during robot motion (online) [13]. While offline methods cannot ensure robot accuracy over long periods of time and require periodic checks, online improvement strategies require costly technological solutions that combine multiple tools (hardware and software) to integrate, process, and use real-time measurements combined with PLC-based compensations.

As mentioned before, the basis for the parametric adjustment is the robot model. Generally, serial industrial robots employ the DH convention improved by non-kinematic parameters such as the elasticity of the joints [14, 15]. The main steps of the calibration process are the referencing of the measuring instrument within the robot's coordinate system, the measurement of multiple robot poses with the reference instrument, the identification of the real kinematic parameters of the robot by iterative minimization of pose error, and the update of the parameters on the robot controller. Many times, the last step is not enabled by robot providers, which means that nominal programs need to be corrected considering the real geometry of the robot.

For dynamic applications, where the robot is subjected to large external forces, other approaches are requested. In these cases, non-kinematic calibration is still susceptible to many improvements. The robot is modeled as an elastic mechanism, and stiffness model relates the configuration of the robot without load and with load or the position with and without load. In this sense, Slavkovic et al. [16] propose to measure the Cartesian stiffness guiding the robot along its XYZ nominal axes guided by an external accessory, and Dumas et al. [17] measure the stiffness of the joint using a free mass. Similarly, Kamali et al. [18] measure joint stiffness, but instead, they apply multidirectional tensile loads via an external attachment.

Another trend that is being addressed in the scientific community to improve the accuracy of a robot with variable process load conditions is the measurement of the pose in real-time to correct its trajectory. This is usually performed by laser tracker or photogrammetric solutions. Although laser-based costly systems are very accurate and provide high-frequency data to close the positioning loop, photogrammetric approaches [19] are much more affordable but neither the accuracy nor the frequency levels of laser trackers can be reached.

The kinematic calculation can be approached as a regression problem. Therefore, machine learning approaches are well-suited for the task, and Morel et al. [20, 21] introduce a method based on support vector regression for computing forward and inverse kinematic calculations.

In the last years, artificial neural networks have proven to achieve exceptional results as function approximations over classical machine learning methods. Chen et al. [22], propose a residual neural network architecture for the regression of non-linear functions. Theofanidis et al [23] used an artificial neural network to model the forward kinematic of a serial robotic system, taking only the TCP position into account but not the orientation. Even if the authors conclude that the classical algebraic method is simple and efficient enough in their setup, their experiment proves that the regression capabilities of neural networks can be used for solving forward kinematics.

Recent studies rely on hybrid models. In these models, the calibration is complemented with machine learning methods designed to minimize the error committed by the robot and, thus, reduce the fork between the accuracy and repeatability of the robot. Chen et al. [24] use a neural network optimized

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

by a genetic algorithm to predict non-linear error on the calibrated robot and a compliance error model to deal with different masses. However, they do not take TCP orientation into account, and no inverse kinematic is proposed. Nguyen et al. [25] and Wang et al. [26] present hybrid kinematic models as a static calibration strategy, in order to compensate for non-geometric errors, but again, only TCP position is taken into account without orientation. To obtain the desired joints, they apply the inverse kinematic for the compensated pose modeling the robot via DH parameters, which may be erroneous since the DH parameters do not offer an accurate modeling.

Inverse kinematics is a computationally and analytically more complex problem in robotics since it is ill-posed for serial manipulators, that is multiple or no solution may exist for the one input. There are also some studies which tackle the inverse kinematic problem via neural networks. In refs. [27, 28], dense artificial neural networks are employed to compute the inverse kinematics of a robot model, the position and rotation values are directly fed to the neural network, and the output is the joint configuration. However, the function approximation behavior of the neural networks is not adequate to deal with multiplicity solutions In ref. [29], a neural network models the forward kinematics, then the network weights are updated by backpropagation and the desired joint values are obtained from the final weights. However, the experiments were only performed using a 3-joint robot in a planar space, which has low dimensionality. Thus, it is uncertain whether the findings can be accurately generalized to the more common scenario of a 6-joint robot. In ref. [30], the current robot joints are also treated as an input for a neural network in order to solve non-triviality, but the augmentation on the input dimensions imposes a higher amount of training data.

The main advantages, compared with other works, is the correction of orientation component which allows accurate control of the end effector in the SE(3). Additionally, we present both forward and inverse kinematic formulations, and our novel inverse kinematic approach can handle multiple solution scenarios.

3. Methodology

The proposed method aims to improve accuracy in the kinematic modeling of robotics systems. A hybrid forward kinematic model is proposed, that integrates traditional mathematical approaches with artificial neural networks. The idea behind this approach is to profit from the strengths of each component, where the mathematical model provides an initial approximation of the pose, while the neural networks predict deviations or errors. By combining these outputs, a more precise and accurate pose estimation can be obtained, see Figure 2 for a schematic representation. The proposed approach uses two separate networks for position and orientation since these measures have different physical spaces. The network input layers are designed to correspond to the robot joints, capturing the joint configuration as input data (q). On the other hand, the output layers of the neural network consist of three neurons. These neurons are responsible for expressing the position correction along the axis $(\Delta x, \Delta y, \Delta z)$ or correction of rotation components $(\Delta\varphi, \Delta\theta, \Delta\psi)$. The estimated position and orientation values provided by the mathematical model are indicated by $\hat{x}, \hat{y}, \hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}$. These estimated values are combined with the corresponding errors resulting in the corrected values denoted as $x^*, y^*, z^*, \varphi^*, \theta^*, \psi^*$.

Indeed, the methodology described requires a calibration process, which can be viewed as the training phase of the neural network. During this calibration process, the neural network is trained as a regression model with the objective of predicting the modeling error associated with each joint configuration

The neural network models are trained on the difference between actual and estimated Cartesian poses, which is computed as:

$$T_{diff}(A, B) = T_A \cdot {T_B}^T \tag{1}$$

where T_A and T_B represent transformation matrixes of the real and estimated poses, respectively. The Cartesian error is not a straightforward metric since it is a combination of rotation and translation errors which have different units and physical meanings. The idea of using two separate networks is that each

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

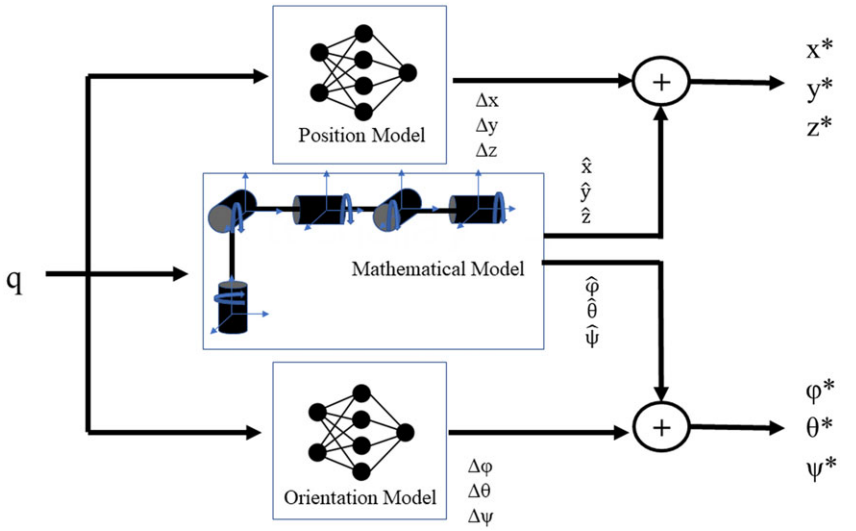


Figure 2. *Scheme of the hybrid forward kinematic model.*

network focuses on a separate physical cost function and minimizes translation and rotation errors independently. The loss function employed in the position model corresponds to the Euclidean norm of the translational part of T_{diff} while the loss function employed for the orientation model corresponds to the norm of the Axis-Angle representation of T_{diff}.

The forward kinematic calculation can be expressed as a mathematical function as it maps a specific set of joint configurations to a corresponding target pose in Cartesian space. However, this is not the case for the inverse kinematic calculation where one target pose in Cartesian space may have multiple possible joint configurations.

To solve the inverse kinematic problem of a hybrid model, the proposed method employs a constrained optimization over a reduced space. First, an approximate solution is extracted from the mathematical model using traditional methods such as [31], these methods search iteratively for a solution that minimized the Cartesian error, starting from a seed. Thus, the seed choice determines which one of the multiple solutions is found.

Subsequently, the approximate solution serves as an initial estimation in a non-linear least squares problem with bounded variables. The choice of a good initial state leverages the optimization process. The variables in the optimization problem are constrained within a small interval around the initial values obtained from the mathematical model. This choice is based on the belief that the actual solution lies in close proximity to the mathematical solution.

The cost function to be minimized corresponds to the Cartesian difference between the target pose and the Cartesian pose reached by the current joint configuration. The variables to be optimized correspond to the joint configuration; therefore, the hybrid forward model is integrated into the optimization process to provide accurate mapping between joints and Cartesian poses, see Figure 3 for a visual scheme.

In this case, it is necessary to express the Cartesian error as a single metric. Consequently, various metrics have been proposed to represent this error, which weighs translation and rotation error in different manners. For example, [32] and [33] use the squared norm of a 6-dimensional vector comprising translational error and orientation error as Euler Vector, Han et al. [34] propose a metric based on the dual quaternion representation, which is used in ref. [35]. In our work, we use the same metric proposed in ref. [36] as it is computationally more efficient, which expresses the Cartesian error as follows:

$$\epsilon_c(A, B) = e^T \cdot (T_A - T_B)^{\circ 2} \cdot e \tag{2}$$

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

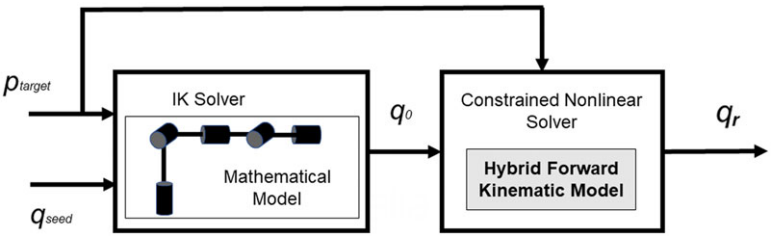


Figure 3. *Scheme of the hybrid inverse kinematic model. p_{target} denotes the target pose in Cartesian space, q_{seed} is initial joint configuration for the classical ik solvers, q_0 describes the computed joint configuration that is used as initial conditions for the second optimization step and q_r refers to resulted joint configuration.*

where T_A and T_B are the rigid transformation matrixes to compare, and e represents a column vector of ones. In this case, translation units are represented in *mm*.

To generate the training data for the neural networks, a dataset consisting of joint positions and corresponding ground truth Cartesian poses is required. The next step involves computing the difference between the ground truth Cartesian poses and the forward pass of the respective joints using the mathematical model.

4. Forward kinematics: experiments and Results

This section describes the performed experiments to evaluate the proposed methodology.

Several models from other studies have been tested (*B1* [23], *B2* [22], *B3* [25], *B4* [24]), together with some defined models based on simple architectures. The defined models are referred as *D1*, *D4*, *D4w*, *D4W*, *Rw,* and *Rd,* where *Dk* stands for a dense neural network with *k* fully connected layers, while suffixes *w* and *W* represent wider and even wider layers, respectively. *Rw* and *Rd* are ResNet Models focused on width and depth correspondingly. A detailed description of these models can be found in the Appendix. Each experiment used different data obtained, from different robots and with different data collection strategies.

4.1. Experiments with virtual data

This experiment uses virtual data to train, evaluate, and test the hybrid model. The objective is to determine which architectures are better suited for this particular task and to ascertain the amount of data required to achieve acceptable performance.

The mathematical model of the hybrid approach corresponds to the calibrated kinematic parameters of a Staubli Robot RX160. In this sense, the ground truth data are collected from a more complex mathematical model which parametrizes joint stiffness. This model accounts for the stiffness of the joints by using linear torsional springs to model their elasticity. The torques exerted on the robot by gravity are calculated using the masses of the links and the end effector. These torques can affect the motion of the robot, which is influenced by the elastic properties of the joints modeled by the torsional springs [37]. In order to train the neural networks, a synthetic data generator was developed using Matlab.

Generated samples are represented by 12-dimensional vectors. The first 6 values are the joint values of the robot, which are used as input for the neural network. The following 6 values correspond to the Cartesian pose difference between mathematical and ground truth values, translational difference is encoded in the first 3 values, and rotational difference is represented by the last 3 values as Axis-Angle.

The main advantage of virtual data is its accessibility, as it is possible to generate big amounts of data. In our experiments, 5,00,000 samples were generated, which cover the joint configuration space in a pseudo uniform distribution. From these samples, 4,00,000 were used for training and the remaining

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

Table I. *Testing results with virtual data.*

Model ID	Average position error (mm)	Average orientation error (degrees)
B1	0.0485	0.0648
B2	0.1796	0.0876
B3	0.2703	0.9496
B4	0.3367	1.1860
D1	0.1502	0.1015
D4	0.0357	0.0565
D4w	0.0226	0.0605
D4W	0.0246	0.0567
Rw	0.0661	**0.0368**
Rd	0.1034	0.0378
Dataset Error	1.8000	0.1700

1,00,000 for validation. Then, 10,000 samples were generated using a random distribution for testing purposes.

4.1.1. Training

In order to facilitate learning of the neural models, min-max normalization is applied to the input of the neural models that is the robot joints are normalized by its full range.

The training was done using the Adagrad Optimizer. It was selected empirically, together with a learning rate of 0.005 and a batch size of 128. The maximum number of epochs is 300 but with a stopping condition if validation accuracy does not present any significant improvement.

Results on the testing dataset are shown in Table I, and the average dataset error is defined as the mean error between the mathematical and ground truth data. The defined dense models obtained better results for the position, but the proposed ResNet models outstand for the orientation. Figure 4 displays distribution comparison of position and orientation errors between the mathematical model and the hybrid model using *D4w* for position and *Rw* for orientation. It can be seen that predicted values tend to have closer to zero errors.

A secondary purpose of this experiment was to evaluate the amount of data needed to obtain the required translation and rotation error. With this aim, models *D4w* and *Rw* were trained and tested with different amounts of data, and compared with the average error of the dataset, see Figure 5. As expected, to some extent the more the data, the better the results. However, even though the improvement for the position is negligible when increasing the data over 100000 cases, it is not the case for the orientation where the error continues to decrease when enlarging the amount of data. As a result, at least 100000 samples are needed in order to obtain an accuracy lower than 0.1 mm in position and 0.1^o in orientation for these particular case, which represent an error reduction of 95% and 42% respectively.

4.2. Experiments with real data using a industrial robot

This experiment was performed with a real industrial robot which is more accurate and repetitive than the collaborative one. A zig-zag trajectory is used for training, and a standard trajectory is used for validation following the ISO 9283 standards, see Figure 6.

The data acquisition process has been conducted by combining a serial robot Staubli RX160 and a portable measuring instrument such as the Leica Absolute Tracker AT960-MR (accuracy of 23 μm). Moreover, to collect orientation information of the planned trajectory, an additional measuring tool, named T-Mac has been used, see Figure 7.

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

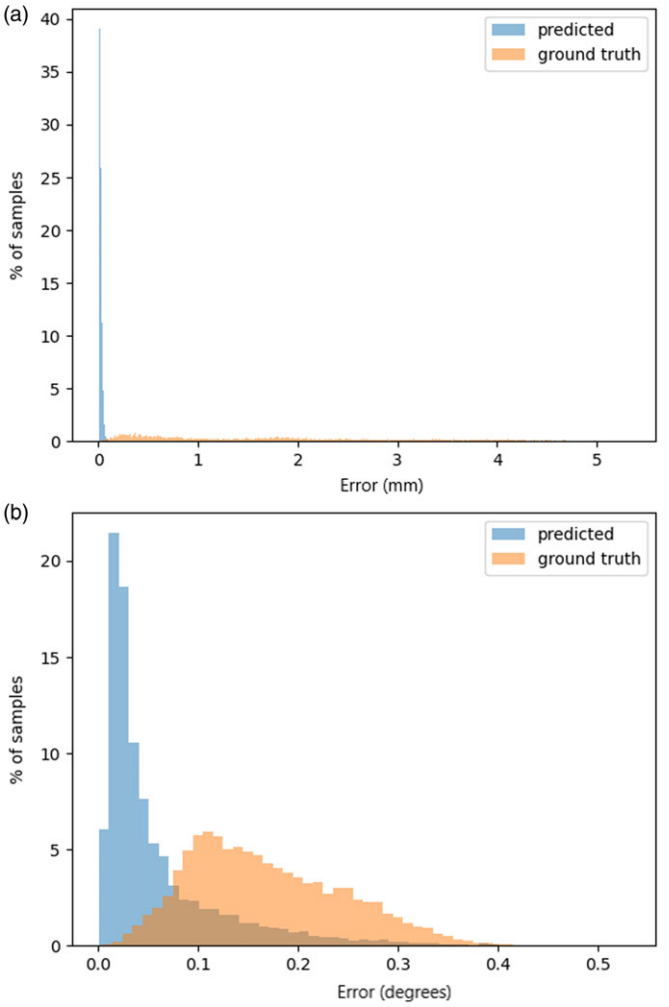


Figure 4. *Distributions of position (a) and rotation (b) Errors achieved with the pure mathematical model (orange) and the hybrid model (blue). Hybrid model using D4w for position Rw for rotation.*

The RX 160 Staubli robot is a 6-axis industrial robot with a nominal load of 20 kg and a reachability of 1600mm. It is considered a high-precision model with a nominal repeatability of ±0.050 mm according to the manufacturer. However after some testing, a real repeatability of ±0.015 mm was observed. Regarding the measuring instrument and technology, a laser tracker and a 6 degrees of freedom device have been gathered to measure the real pose of the robot for each of the points defining the different trajectories. In order to measure the pose of the robot, a referencing process has been performed to characterize the spatial relationship between the laser tracker and the robot base as well as the T-Mac and the robot flange. The referencing method corresponds to ref. [38], and it involves moving joints 1 and 2 independently while recording their positions with a laser tracker. These recorded points are then used to fit a plane, determining the normal vector representing the rotation axis. The origin of the frame is defined as the intersection point of this rotation axis with the base plane, also measured with the laser tracker. The same process is repeated for the tool-frame transformation, this time utilizing joints 5 and 6. Once all transformations are known, the raw position and orientation measurements can be directly converted to the robot base reference.

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

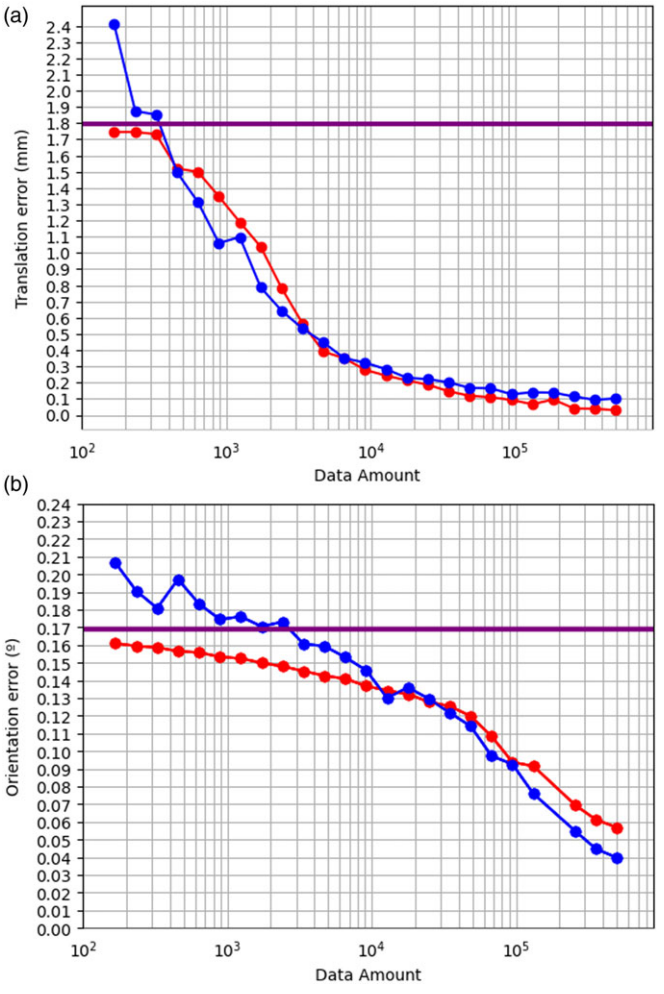


Figure 5. *Relation between accuracy and training data quantity for model D4w (red) and model Rw (blue). The purple horizontal line represents the average error of the training dataset.*

Regarding the measurement process, a dynamic strategy was discarded to avoid dynamical error sources not included in the physical model. Moreover, dynamic characterization demands precise synchronization; therefore, triggering between the robot controller and the measuring system is requested, which is not available.

Instead, a static data collection method was selected. In this case, the robot stops for each of the trajectory points and the laser tracker measures the points aided by the stable probing mode. This measuring profile ensures the unequivocal correspondence between robot and laser tracker data and therefore a higher reliability/quality of recorded data. One notable limitation of this approach is that it requires a longer time.

The data points correspond to an ISO9283 cube where there is a small change of orientation along the trajectory. Since the data acquisition method is quite tedious, it took around 16 h, the number of data is reduced compared with the previous experiment. Overall, 12,004 samples of data were collected, and they were shuffled and separated into 3 distinct groups, 8642 are used for training, 2160 for validation, and 1202 for testing. Training is performed as in section 4.1.

In this case, the joint normalization was not applied in the full joint range but in a reduced range regarding the characterized trajectories. The reduced range is obtained from the maximum and minimum

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

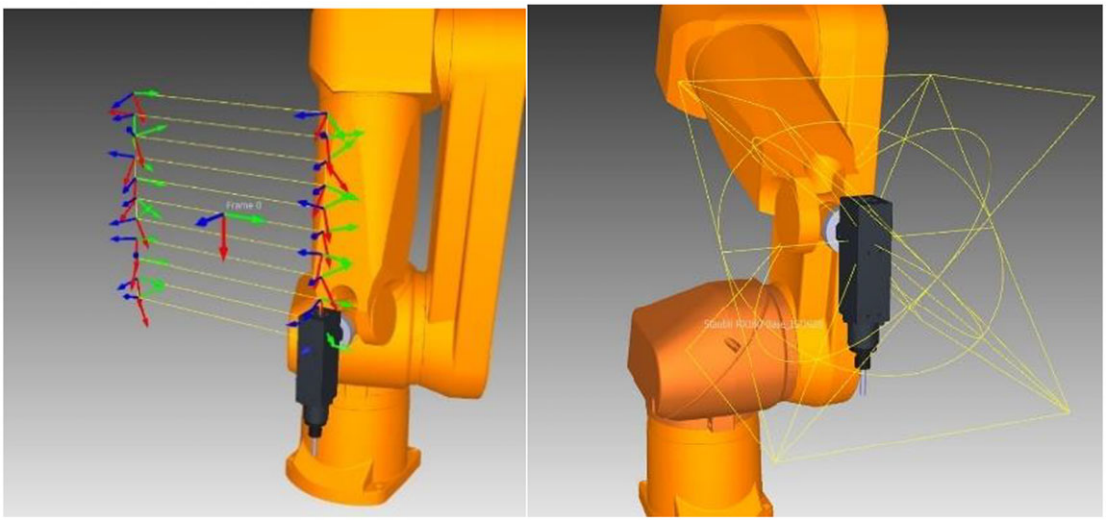


Figure 6. *Trajectories used in the industrial experiment. Left: training trajectory. Right: validation trajectory.*

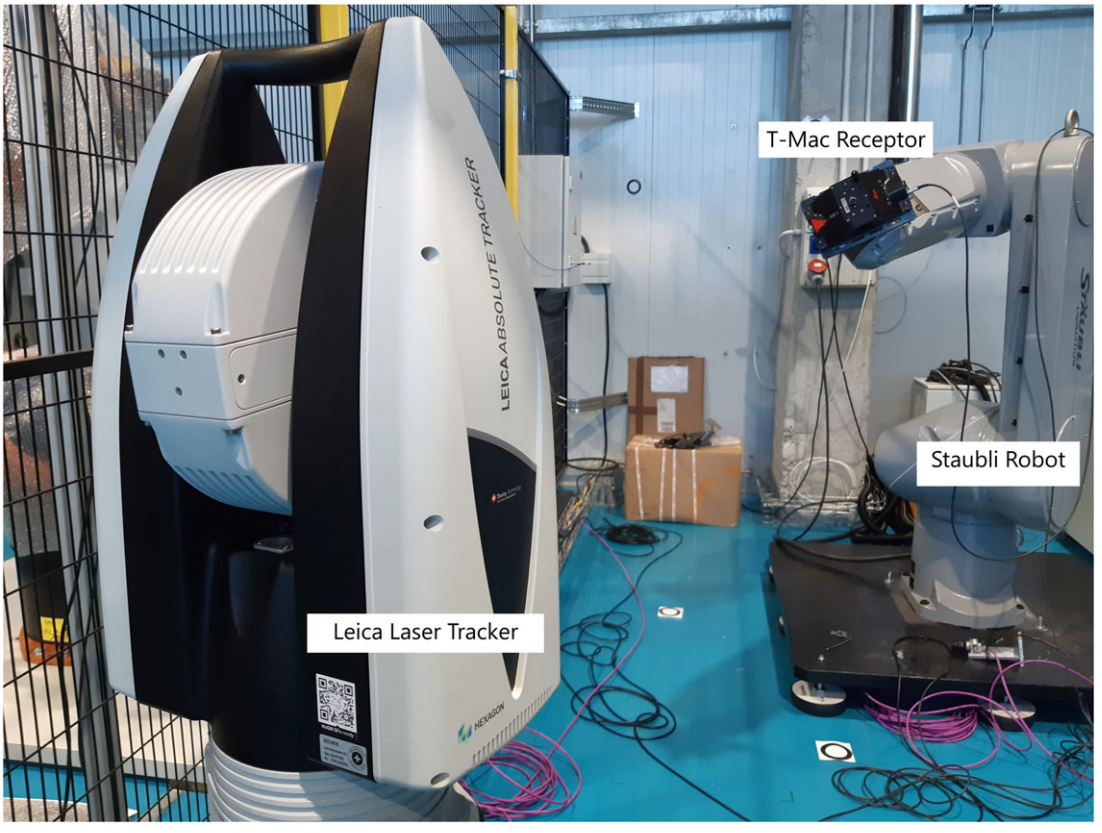


Figure 7. *Data collection physical setup of the industrial case.*

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

Table II. *Testing position and orientation results with industrial robot data.*

Model ID	Average position error (mm)	Average orientation error (degrees)
B1	0.0420	0.0099
B2	0.0348	0.0158
B3	0.0432	0.0155
B4	0.0431	0.0153
D1	0.043	0.0112
D4	0.0419	0.0099
D4w	0.0419	**0.0098**
D4W	0.0419	**0.0098**
RW	**0.0301**	0.0293
RD	0.0328	0.0135
Dataset Error	0.0430	0.0150

values of the joints along the expected trajectories. Therefore, the models are not expected to work with values outside this range since machine learning models do not perform well on extrapolating values outside trained distributions.

Table II presents a summary of the obtained errors for each trained model. In this case, the dataset error presents a mean value of 0.043 mm for position and 0.015 degrees for orientation. Thus, for the position case, the models based on a fully connected architecture (*B1, B3, B4, D1, D4, D4w, D4W*) offer a minimal improvement, while ResNet models (*B2, Rw, Rd*) offer better performance, whereas, for the orientation case, the opposite behavior occurs, the fully connected models achieve smaller error. Figure 8 shows the error distribution of models with different architectures for position and orientation. It can be seen that the position error with the dense model hardly improves, but the orientation error does. On the other hand, with the ResNet model, there is an improvement in the position error and a less pronounced improvement in rotation error, although for the rotation case, the presence of outliers is detected.

The main insight of this experiment is that for a relatively small number of data, the ResNet models offer better performance for the position case while dense models perform better for the orientation case. This coincides with the data quantity analysis from the virtual case, see Figure 5. However, the overall error reduction is minor due to the low number of training data.

4.3. Experiments with real data using a collaborative robot

In this experiment, a collaborative robot is used, whose model does not present as much accuracy as the industrial robot previously considered. The hybrid approach is evaluated with a UR10 from Universal Robots, a 6-axis robot arm with a repeatability of ± 0.1 mm. To capture the position and orientation data of the end effector, a combination of equipment was utilized. The data were obtained using an API Laser Tracker Radian Pro, known for its high accuracy of 15 μm. Additionally, a Smart Track Sensor (STS) was mounted on the robot, providing position and orientation values. To accommodate the working limitations of the STS, a support base was constructed and incorporated into the setup. This base ensured close to vertical alignment of the sensor during data acquisition. The complete setup, represented in Figure 9, enables the acquisition of highly precise data, surpassing the accuracy standards available in the market.

In order to perform the data acquisition, two separate trajectories were defined for training and testing data respectively, both describing an arbitrary movement inside the same working area and without interfering with the laser beam. The Cartesian poses were recorded in a continuous manner. To minimize undesired dynamic behaviors and ensure accurate data collection, the trajectories were performed at low velocities. Due to the disparity in frequencies between the laser tracker (3 kHz) and the UR10

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

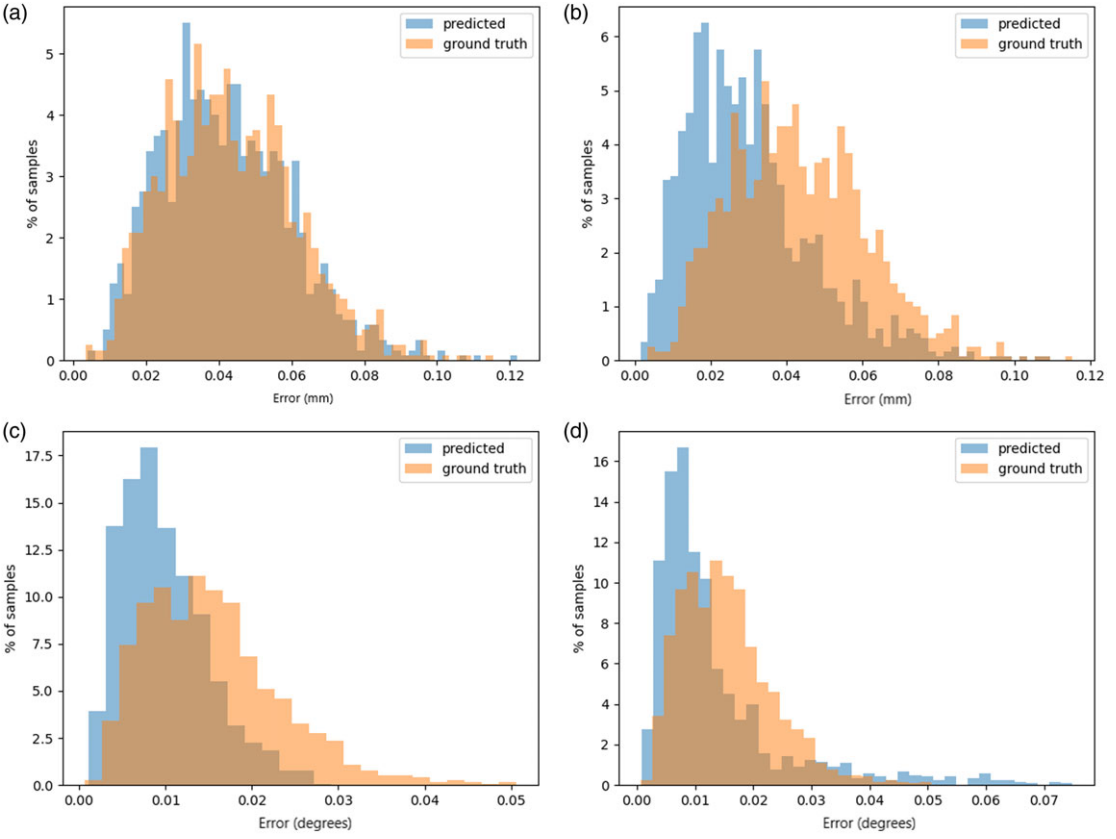


Figure 8. *Distribution of the ground truth error (orange) and the predicted error (blue). Top row (a,b) corresponds to position errors and the bottom row (c,d) to orientation errors. The left column (a,c) corresponds to model D4w and the right column (b,d) to model Rw.*

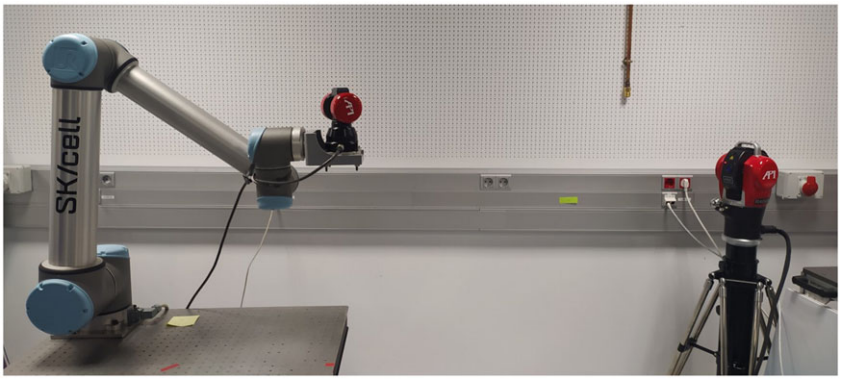


Figure 9. *Measuring setup of the collaborative case. (left) The robot holding the smart tracker sensor and the laser tracker device (right).*

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

Table III. *Testing position and orientation results with collaborative robot data.*

Model ID	Average position error (mm)	Average orientation error (degrees)
B1 [23]	0.3734	0.1162
B2 [22]	0.4525	0.1383
B3 [25]	0.6218	6.6610
B4 [24]	0.8509	6.6480
D1	0.5039	0.1092
D4	0.3766	0.1119
D4w	**0.3620**	0.1175
D4W	0.3839	0.1141
Rw	0.4055	**0.1026**
Rd	0.4176	0.1163
Dataset Error	5.963	2.489

robot (110 Hz), achieving good synchronization of the captured data is feasible. The higher frequency of the laser tracker allows for more frequent data sampling, ensuring that the position and orientation measurements are captured at a finer temporal resolution. Despite the lower frequency of the UR10 robot, the timing differences between the two systems can be effectively managed, enabling accurate data synchronization during the experiment. This synchronization ensures that the captured data from both devices correspond to the same temporal instance, facilitating precise analysis and evaluation of the robot's performance. In this way, train data include almost 250.000 poses acquired during 8 h, while test data consisted of 27.500 poses obtained in less than 2 h.

Ground truth data from the laser tracker are first referenced to the robot base following the method explained in section 4.2. Afterward, each pose is compared with the UR10 calibrated output. This difference, with the joints normalized in a reduced range and presented in the same conditions as described in experiment IV-A, serves to train and test the models. Taking into account the separate position and orientation models, Table III resumes the results for each trained model, and Figure 10 shows the results for model 7 and model 9 in position and orientation, respectively.

The dataset average error is 5.963 mm in position and 2.489 degrees in orientation. This experiment shows that the hybrid approach reduces these distances considerably, for example down to 0.362 mm and 0.1175 degrees with model *D4w* and model *Rw,* respectively. In this case, the number of training data was high enough to appreciate accuracy improvement.

5. Inverse kinematics: experiments and results

To assess the effectiveness of the proposed inverse kinematic model, the hybrid forward kinematic models trained in the previous experiments are used. Thus, the purpose of this experiment is to verify whether a desired Cartesian pose can be achieved by obtaining a corresponding joint configuration using the hybrid forward kinematic model. These experiments were solely conducted for virtual and collaborative scenarios.

5.1. Virtual scenario

Related to the virtual case of section 4.1. 5000 joint configurations were randomly selected from within the joint limit of the robot, and their corresponding Cartesian poses were obtained via the hybrid forward model (model *D4w* for position and **Rw** for orientation). Then, the resulting Cartesian poses were fed to the hybrid inverse model, which outputs the proposed joint configurations. From these suggested joint configurations, their corresponding Cartesian poses were extracted using again the hybrid forward

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

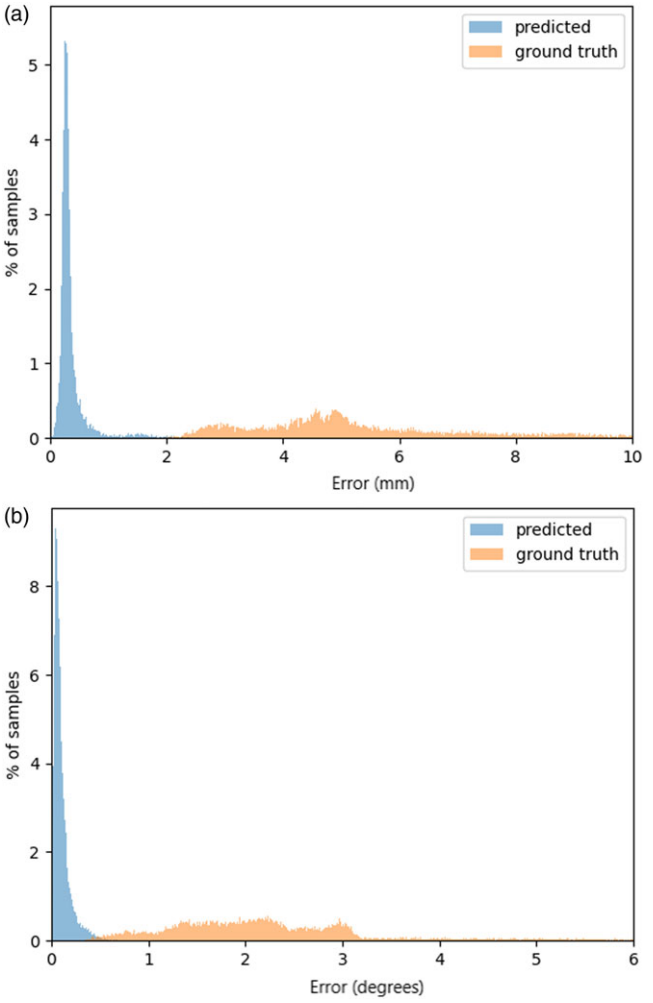


Figure 10. *Position and orientation error, real(orange) denotes the error of the mathematical model, and predicted (blue) is the error of the hybrid model with respect to the ground truth.*

model. Finally, position and orientation errors between the target and current Cartesian poses are computed. Our method was compared against an adapted version of the one proposed by Wang et al. [26]. They propose a method where an initial approximate solution is obtained from a pure mathematical inverse kinematic solver and fed to the hybrid forward model. The hybrid forward model then generates a corrected Cartesian pose by considering any deviations or errors predicted by the neural network component. This corrected Cartesian pose is then fed back once into the mathematical solver to further extract the joint configuration. The adaptation consisted of using our hybrid model as forward model, therefore orientation is also considered.

Table IV shows position and orientation error statistics, and we believe that these metrics are more intuitive to the reader than the cost from equation 2. Our method achieves near to zero error and performs better when compared to the proposed in ref. [26] whose error is substantial for an accuracy model.

5.2. UR10 collaborative robot scenario

In the case of the collaborative robot, taking into account data from the testing trajectory, tracked and processed results are used. Setting these like target poses, the inverse kinematic is applied to obtain the

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

Table IV. *Inverse kinematic error for the hybrid model trained with virtual data.*

	Translation error (mm)		Orientation error (degrees)	
	mean	**std**	**mean**	**std**
Our method	6.02e-6	1.25e-4	4.85e-4	0.011
Method in [26]	4.52	0.078	0.397	1.13

Table V. *Inverse kinematic error for the hybrid model trained with collaborative data.*

	Translation error (mm)		Orientation error (degrees)	
	mean	**std**	**mean**	**std**
Our method	0.0012	0.0011	1.38	1.71
Method in [26]	8.89	3.2	4.03	2.11

predicted joint configurations. These are used with the hybrid model to calculate the forward kinematics and obtain a predicted pose.

Using an arbitrary selection of 1000 of these test points, results are measured by comparing the predicted pose with respect to the real target. Table V presents the results both in position and orientation. The position error of our model is very low whereas the orientation error can be improved.

In summary, our method achieves very low error thanks to the use of the hybrid forward kinematic model as feedback. The method in ref. [26] has a higher error as it only uses the hybrid forward kinematic to compute a compensated target pose but the inverse kinematics only takes into consideration the mathematical model and it was only conceived for position and not orientation.

6. Discussion

The performed experiments have demonstrated the effectiveness of the proposed hybrid kinematic model in improving accuracy. The model was able to reduce the Cartesian error based on ground truth data, indicating its ability to provide more accurate results compared to state-of-the-art models.

To evaluate the performance of the inverse kinematic model, it was tested separately and compared with the forward kinematic solution. The positive result obtained suggests that the inverse kinematic model is reliable and can complement the overall accuracy of the hybrid kinematic model.

The choice of the neural network model is crucial, and the experiments conducted in this study provide insights into selecting the appropriate architecture. When the data amount is relatively low (less than 10,000 samples), ResNet architectures performed better for position, while dense architectures showed better performance for orientation. However, as the data amount increased, this behavior was reversed. This suggests that the selection of the neural network architecture should be based on the available data and its quantity.

As expected for a data-driven method, the number of training data plays a significant role in the resulting accuracy of the model. In this work, we trained the models with a higher number of samples in comparison with the rest of literature reviewed. This approach aims to enhance the generalization capabilities of neural networks. With a larger dataset, the neural network models can have more parameters, allowing them to approximate more complex functions and achieve better performance.

In the virtual and collaborative cases, where we have a higher data amount, the best performance models were *D4w* and *Rw* for position and orientation respectively.

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

However, the process of obtaining such amount of data is not an easy task. In the experiments, two data collection techniques were explored, and the recommended technique depends on the robot type, controller connection options, or the requested accuracy. Dynamic data collection is faster but it is more complex due to the requirement of a synchronization protocol. For the case of collaborative robots whose precision margin is high, a protocol based on frequencies can be sufficient as shown in experiment 4.3. Notwithstanding, if high precision is demanded, a triggered-based synchronization protocol is needed. Static data collection is easier to implement since it does not require any additional devices and it ensures point correspondence but it is excessively time-consuming.

The ground truth data obtained from the laser tracker offer exceptional accuracy. However, when transforming the measurements from the laser tracker coordinate system to the robot coordinate system, a referencing procedure is needed. This procedure may introduce small errors that may propagate throughout the data collection process.

Certainly, the proposed method has a few limitations. Firstly, it requires a high amount of quality data, which necessitates both time and hardware resources. This implies that collecting sufficient data for the experiment can be a time-consuming process, and the hardware requirements may be substantial.

Additionally, the experiment imposes hardware limitations on the trajectories. These limitations involve specific criteria such as the measurement sensor requiring some verticality and occlusion avoidance from the laser tracker perspective. These constraints restrict the freedom of movement of the robots and may limit the practicality of the method outside of the predefined working space.

Another limitation is related to the measurement tools being mounted on the robot flange. This means that the obtained results may only be replicable with tools of similar mass or inertias. If different tools or equipment are used, it could introduce variations in the results, making them less comparable or applicable in different scenarios.

Despite these inconveniences, the laser tracker is considered the most suitable option due to its high accuracy and measuring frequency. The use of a laser tracker can provide precise measurements, enabling accurate analysis and evaluation of the robot's performance within the predefined working space.

The presented methodology can be classified as an offline static non-parametric calibration strategy. It involves the acquisition of data and subsequent model training procedures to improve the accuracy of the robot's kinematic model. It is important to note that this calibration methodology is specifically designed for high-accuracy scenarios. Its purpose is to enhance the accuracy of the robot's kinematic model beyond what can be achieved with classical kinematic approaches. It does not aim to substitute classical kinematic approaches as they are simpler, faster, and provide reasonable accuracy for a wide range of tasks, but it can be thought of as a complementary approach.

7. Conclusion and future work

This work presents a hybrid kinematic modeling formulation to achieve accurate positioning of the tool of a serial robotic manipulator. Our formulation is hybrid in the sense that it relies on classical analytical formulation and data-driven methods.

The main contribution of the presented work to the state-of-the-art lies in the development of a hybrid architecture for both forward and inverse kinematic calculations. What sets this work apart is the inclusion of the orientation component, which distinguishes it from other hybrid approaches found in the literature. To enable this hybrid architecture, a calibration procedure is implemented, which involves data collection and model training for each individual robot. This calibration step ensures that the kinematic model accurately represents the specific characteristics and behavior of each robot. By tailoring the model to the individual robot, it enhances the accuracy and precision of the kinematic calculations at the expense of a more complex calibration process.

Experimentation shows that the hybrid model formulation is able to improve accuracy when compared to calibrated kinematic mathematical models.

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

Our experiments involving real robots rely on laser tracker systems to obtain ground truth data, so certain tools were attached to the robot for laser reflection for the data collection process. Tool mass and inertia may contribute to the positioning precision of the robot. Therefore, our future extensions on this work will focus on studying the end effector masses and inertia effects and the integration of the mentioned model on a real robot.

Author contributions. MO, AE, and GK conceived and designed the study, MO, AE, GK, BA, and JF conducted data gathering, MO and JH involved in statistical analyses, MO, EL, and XL wrote the article.

Financial support. This work was supported by Elkartek 2021 funding program under Grant 0124/2021

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

References

[1] M. Aiman, K. Bahrin, F. Othman, N. Hayati, N. Azli and F. Talib, "Industry 4.0: A review on industrial automation and robotics," *J Teknol* **78**, 2180–3722 (2016).
[2] B. Robert, "The growing use of robots by the aerospace industry," *Ind Robot An Int J* **45**(6), 705–709 (2018).
[3] A. Hassan, A. El-Habrouk, M. Deghedie and S. Deghedie, "Renewable energy for robots and robots for renewable energy - a review," *Robotica* **38**, 1–29 (2019).
[4] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew and F. Makedon, "Survey of robots in healthcare," *Technologies* **9**(1), 8 (2021).
[5] P. Urhal, A. Weightman, C. Diver and P. Bartolo, "Robot assisted additive manufacturing: A review," *Robot Comp Int Manuf* **59**, 335–345 (2019).
[6] W. Veitschegger and C. H. Wu, "Robot accuracy analysis based on kinematics," *IEEE J Robot Autom* **2**(3), 171–179 (1986).
[7] J. Motta, Calibration: Modeling measurement and applications, (2006).
[8] A. Sirinterlikci, M. Tiryakioglu, A. Bird, A. Harris and K. Kweder, "Repeatability and accuracy of an industrial robot: Laboratory experience for a design of experiments course," *Technol Interf J* **9** (2009).
[9] R. Bernard, *Calibration First Edition* (Springer, US, 1993).
[10] L. Ginani and J. Motta, "Theoretical and practical aspects of robot calibration with experimental verification," *J Braz Soc Mech Sci Eng* **33**(1), 15–21 (2011).
[11] Z. Roth, B. Mooring and B. Ravani, "An overview of robot calibration," *IEEE J Robot Autom* **3**(5), 377–385 (1987).
[12] A. Brahmia, A. Kerboua, R. Kelaiaia and A. Latreche, "Tolerance synthesis of delta-like parallel robots using a nonlinear optimisation method," *Appl Sci* **13**, 10703 (2023).
[13] X. Shi, F. Zhang, X. Qu and B. Liu, "An online real-time path compensation system for industrial robots based on laser tracker," *Int J Adv Robot Syst* **13**(5), 172988141666336 (2016).
[14] M. Vincze, S. Spiess, M. Parotidis and M. Götz, "Automatic generation of nonredundant and complete models for geometric and non-geometric errors of robots," *Int J Model Simul* **19**(3), 236–243 (1999).
[15] T. Braun, *Embedded Robotic: Robot Manipulators* (Springer, Singapore, 2022) pp. 253–269.
[16] N. Slavkovic, D. Milutinovic, B. Kokotovic, M. M. Glavonjic, S. Zivanovic and K. F. Ehmann, " Cartesian compliance identification and analysis of an articulated machining robot," *FME Trans* **41**, 83–95 (2013).
[17] C. Dumas, S. Caro, S. Garnier and B. Furet, "Joint stiffness identification of industrial serial robots," *Robot Comp Integ Manuf* **27**(4), 881–888 (2011).
[18] K. Kamali, A. Joubair, I. A. Bonev and P. Bigras, "Elasto-Geometrical Calibration of an Industrial Robot Under Multidirectional External Loads using a Laser Tracker," **In:** *IEEE International Conference on Robotics and Automation (ICRA)*, (2016) pp. 4320–4327.
[19] A. Mendikute, J. A. Yagüe-Fabra, M. zatarain, A. Bertelsen and I. Leizea, "Selfcalibrated in-process photogrammetry for large raw part measurement and alignment before machining," *Sensors* **17**(9), 2066 (2017).
[20] A. Morell, M. Tarokh and L. Acosta, "Inverse Kinematics Solutions for Serial Robots Using Support Vector Regression," **In:** *IEEE International Conference on Robotics and Automation*, (2013) pp. 4203–4208.
[21] A. Morell, M. Tarokh and L. Acosta, "Solving the forward kinematics problem in parallel robots using support vector regression," *Eng Appl Artif Intel* **26**(7), 1698–1706 (2013).
[22] D. Chen, F. Hu, G. Nian and T. Yang, "Deep residual learning for nonlinear regression," *Entropy* **22**(2), 193 (2020).
[23] M. Theofanidis, S. I. Sayed, J. Cloud, J. Brady and F. Makedon, "Kinematic Estimation with Neural Networks for Robotic Manipulators," **In:** *International Conference on Artificial Neural Networks*, (2018) pp. 795–802.
[24] X. Chen, Q. Zhang, Y. Sun and P. Crippa, "Evolutionary robot calibration and nonlinear compensation methodology based on GA-DNN and an extra compliance error model," *Math Prob Eng* **2020**, 1–15 (2020).

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

[25] H. N. Nguyen, J. Zhou and H. J. Kang, "A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network," *Neurocomputing* **151**(3), 996–1005 (2015).
[26] Z. Wang, Z. Chen, Y. Wang, C. Mao and Q. Hang, "A robot calibration method based on joint angle division and an artificial neural network," *Math Probl Eng* **2019**, 1–12 (2019).
[27] S. K. Shah, R. Mishra and L. S. Ray, "Solution and validation of inverse kinematics using deep artificial neural network," *Mater Today Proc* **26**, 1250–1254 (2020).
[28] A.-V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Proc Technol* **12**, 20–27 (2014).
[29] P. Srisuk, A. Sento and Y. Kitjaidure, "Inverse Kinematics Solution Using Neural Networks from Forward Kinematics Equations," **In:** *9th International Conference on Knowledge and Smart Technology (KST)*, (2017) pp. 61–65.
[30] A. R. Amusawi, L. C. Dülger and S. Kapucu, "A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242)," *Comput Intel Neurosc* **2016**, 1–10 (2016).
[31] P. Costa, J. Lima, A. Pereira and A. Pinto, "An Optimization Approach for the Inverse Kinematics of a Highly Redundant Robot," **In:** *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA*, (2015) pp. 433–442.
[32] H. Bruyninckx, P. Soetens and B. Koninckx, "The Real-Time Motion Control Core of the Orocos Project," **In:** *IEEE International Conference on Robotics and Automation*, (2003) pp. 2766–2771.
[33] P. Corke, "Robotics, vision and control - fundamental algorithms in MATLAB," *Springer Tracts Adv Robot* **73**, 1–495 (2011).
[34] D.-P. Han, Q. Wei and Z.-X. Li, "Kinematic control of free rigid bodies using dual quaternions," *Int J Autom Comp* **5**(3), 319–324 (2008).
[35] P. Beeson and B. Ames, "Trac-ik: An Open-Source Library for Improved Solving of Generic Inverse Kinematics," **In:** *IEEE RAS Humanoids Conference* (2015) pp. 928–935.
[36] P. Maceron, "IKPy," doi: 10.5281/zenodo.6551158 (2015).
[37] A. Nubiola and I. A. Bonev, "Absolute calibration of an ABB IRB. 1600 robot using a laser tracker," *Robot Com Int Manuf* **29**(1), 236–245 (2013).
[38] L. Lattanzi, C. Cristalli, D. Massa, S. Boria, P. Lepine and M. Pellicciari, "Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task," *Int J Adv Manuf Technol* **111**(7), 1813–1829 (2020).

Appendix

The defined models presented in this work are described in this section. These models can be clustered in two architecture types, models *D1, D4, D4w,* and *D4W* are fully connected models whereas models *Rw* and *Rd* present a ResNet architecture.

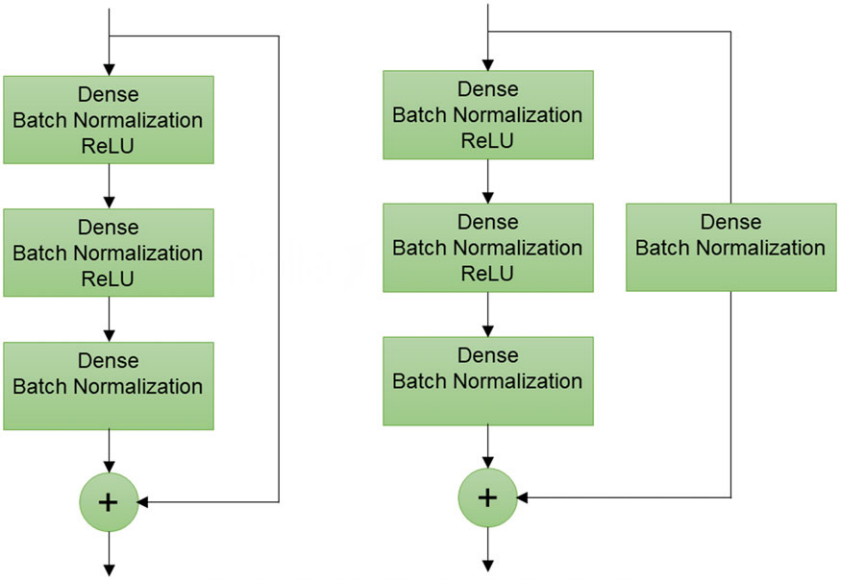


Figure 11. *Diagram of the ResNet blocks used for models Rw and Rd.*

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press

All fully connected models present ReLu as activation functions in each layer except for the output layer which has a linear activation function. They can be summarized as follows:

- Model *D1* presents 1 fully connected layer with 5000 neurons.
- Model *D4* presents 4 fully connected layers with the following number of neurons 3500-2500-1500-750, respectively.
- Model *D4w* presents 4 fully connected layers with the following number of neurons 5000-3000-1000-500, respectively.
- Model *D4W* presents 4 fully connected layers with the following number of neurons 6000-3000-1500-1000, respectively.

The ResNet models are composed of the identity and dense block presented in ref. [22], see Figure 1; these blocks are grouped in components, which are composed of one dense block followed by two identity blocks with the same number of neurons. They can be summarized as follows:

- Model *Rw* presents 3 components with widths of 700,500, and 200.
- Model *Rd* presents 5 components with widths of 500,200,100,70, and 30.

Cite this article: M. Ojer, A. Etxezarreta, G. Kortaberria, B. Ahmed, J. Flores, J. Hernandez, E. Lazkano and X. Lin (2024). "High accuracy hybrid kinematic modeling for serial robotic manipulators", Robotica **42**, 3211–3229. https://doi.org/10.1017/S026357472400136X

https://doi.org/10.1017/S026357472400136X Published online by Cambridge University Press