Hybrid calibration for robot full pose accuracy enhancement using active learning based method

Maxime Selingue, Stéphane Thiery, Adel Olabi and Richard Béarée, *Senior Member, IEEE,*
LISPEN, *Arts et Métiers Institute of Technology*
Lille, France
Corresponding author : maxime.selingue@ensam.eu

***Abstract*—Calibration processes came up to face the growing demand for robot's accuracy in the industry. Calibration methods model the physical phenomena that degrades the accuracy, identify those model's parameters through measurement and compensate them. The most efficient calibration methods combine analytical models with machine learning, thus are called hybrid. However, the most commonly used machine learning tool for hybrid calibration, i.e. artificial neural network, are data consuming, leading to long and complex measurement processes. In this paper, a new active learning based hybrid calibration method is presented. A fine tuning of the method parameters have been made on a fully measured dataset, and an accurate stopping condition has been found. Thus, the method is data-efficient and provides better results that the ANN-based hybrid calibration methods. Experimental validation shows that with this method, positioning error could be reduced by 90% on two different robots, using only a few data. The method is able to reduce orientation error by 95% as well. Data-efficiency makes the measurement process faster than ANN-based hybrid calibration, thus avoiding thermal effects on the robot, which lead to more accurate measurements.**

***Index Terms*—Industrial robot accuracy, Hybrid Calibration, Robotics, Machine Learning**

I. Introduction

The use of robots in the industry is constantly growing, with wide range of activities from traditional pick-and-place application to more complex processes such as machining, drilling, welding etc. Those tasks require high accuracy. Since industrial robots are highly repeatable, but have a poor absolute accuracy, they are often programmed online, with teaching methods, for high accuracy demanding tasks. However, offline programming is more and more used in the industry, as the demand for digital twins grew up. Thus, the deviation between the actually reached cartesian poses (i.e. position and orientation) of the end-effector (EE) of a robot and the expected one has to be reduced. This deviation is caused by various phenomena. First, robot's geometric parameters (joints orientation, links lengths etc.), that are used to compute forward and inverse kinematics, suffer from manufacturing and assembly tolerances. In addition, other non-linear phenomena, such as deflection, backlash or encoder resolution, can affect the absolute accuracy of industrial robots [1]. To overcome this issue, robot calibration must be performed. Traditionally, one can distinguish two types of calibration: model-based and model-less.

Model-based calibration relies on an accurate analytical model of the phenomena that reduce a robot's accuracy. The most known *model-based* calibration method is the geometric calibration, which is based on the identification of the geometric parameters (i.e. position and orientation of each joint with regard to the previous one) [2]–[4]. The efficiency of geometric calibration can be enhanced using observability indices, opening the way towards optimal experiment design [5], [6]. Moreover, the stiffness of each axis can be identified as well, so that the self-mass and the payload of the robot can be compensated for. In [7], Salisbury first established the kinematic model that involve axis stiffness. Later, authors in [8] proposed a method to identify axis stiffness, knowing the mass model of the robot (i.e. the mass and the center of gravity of each link). However, since the mass model of industrial robot is rarely available, some other methods were proposed in the literature to estimate axis stiffness for payload compensation only. In [9], a method was proposed to identify the stiffness of all axis knowing the wrench vector applied to the robot's EE. In [10], the axis stiffness of an industrial robot were identified one axis after the other. Later, in [11], authors identified a collaborative robot's axis stiffness using the torque sensors of the robot. However, all these previous methods have drawbacks. Some of them require either to know the mass model or the robot to be equipped with torque sensor, which is rarely the case. Other methods estimating axis stiffness from the wrench applied does not take into account the self-mass of the robot. This can be an issue because axis stiffness mainly comes from gearboxes in each axis, whose stiffness is not constant: it increases as the applied torque increases, according to [12]. This can be taken into account in the stiffness formulation, but this leads to a very complex model. In [13], joint stiffness is modeled by a second order polynomial function of the torque applied, thus three coefficients need to be identified for each axis. Similarly to the observability indices that can enhance geometric calibration processes, [14], [15] propose a method to optimize the measurement process for stiffness identification as well. Later, in [16], authors proposed a method for estimating geometrical parameters and joint stiffness simultaneously with an optimized measurement set.

Model-based calibration can compensate for most of the positioning error through geometric parameters and axis stiffness identification in an understandable way (using analytical models), however the models grow in complexity when the accuracy requirements are high. Extensively, some phenomena are too complex to be compensated for using analytical models

(e.g. gearboxes irregularities, thermal effects). To overcome this issue, *model-less* calibration is based on machine learning models that are to approximate the positioning error of the robot's EE. Most of calibration methods for static positioning use Artificial Neural Network (ANN), since it is an universal regression tool. In [17], authors show in simulation that ANNs were able to compensate for kinematics error of a 2 degrees of freedom robot. In [18], a method based on an ANN is proposed. The ANN predicts the angular offsets to apply to each joint in function of the desired cartesian position. The method has been tested in simulation and showed a significant reduction of the positioning error. In [19], authors show in simulation that only a few amount of data is required to train an ANN for error compensation of a two-axis robot in a small area. In [20], a BP-neural networks is used to directly predict the position of the end-effector given a joint position for a robotic poly-articulated arm coordinate measuring machine.

In the past years, *hybrid* calibration (combining both *model-based* and *model-less* calibration) showed interesting results. In [21], authors identify the geometric parameters of a PA-10 robot and used an ANN to predict the angular offset to apply given a joint position, so that the desired cartesian position is reached accurately. In [22], [23], authors use an ANN to predict the difference between real measurements and a kinematic model of the robot with identified geometric parameters and axis stiffness. The method have been tested on a HH800 robot to compensate for only self-mass of the robot (i.e. not the payload). In [24], authors train an ANN for residual error prediction using massive measurements, made possible by the automation of the measurement process with a trajectory that keep the contact between the laser tracker and the Spherical Mounted Reflector (SMR). In [25], authors improve the positioning and the orientation accuracy of an industrial robot using ANN and geometric parameters identification across the whole workspace of the robot, using almost 14000 data that needed 3.5 days to be measured. In [26], an hybrid calibration method also based on geometric parameters identification and artificial neural network is proposed. The ANN used takes as input the joint configuration and the payload as well, so that any new payload can be compensated for. Consequently, the ANN's training requires measurement with different payloads, which leads to complex measurement process. Since ANN are known to be data-consuming, while measurement are costly because of the measurement devices used, some researches tends to reduce the number of data required, while keeping a high accuracy. Indeed, in [27], authors show that hybrid calibration can reduce the positioning error for a 5-bar parallel robot. They optimize the training phase of the neural network to reduce the number of data required using a Levenberg-Marquardt combined with accelerated particule swarm for weights optimization.

However, some researches showed that Gaussian Process Regressors (GPRs) can be as effective as ANN's for robot calibration. In [28], authors used GPR to approximate the positioning error of the end-effector (EE) of the robot. In [29], [30], GPR are used to model more accurately robot's dynamics. These methods were based on GPR instead of ANN and obtained satisfying accuracy, but some of the GPR advantages over ANN are not exploited, such as measurement's noise modeling and uncertainty indicator over predictions. In [31], an active learning algorithm using the variance output of a GPR is proposed and tested on open-source datasets in other contexts than robotic calibration.

Our previous work [32] focused on experimental studies to reduce the number of data needed for ANN's training, in scenarios where the robot handle heavy payloads. However, even though this work showed that ANN's coupled with geometric calibration requires significantly less data than model-free calibration, there is currently no mean to know how many data are sufficient. One have to get too much data before knowing that less data would have been enough, leading to non-optimal measurement processes. Moreover, ANN can handle any regression problem, given they have enough neurons, nonetheless they are not necessarily the most efficient tool. Indeed, GPR are known to be more effective in low-dimension and low-data contexts, which is the case in robot calibration. However, methods using GPR for robotic calibration does not exploit their full potential, as they are only used for regression purposes but not for optimizing the measurement process as well.

For both *model-less* and *hybrid* calibration, since model's training often requires a lot of data, the measurement processes are time-consuming. Indeed, in [24], authors stated that the measurement process lasted for 5 hours to measure more than 10 000 points. In our previous work [32], the measurement process took 3.5h to acquire 2000 datas. Thus, during the measurement process, motors temperature increases, causing thermal drift that bias the dataset. In [13], authors experimentally observed that the stabilization of the joint's temperature of an aluminum-made robot appears after 2.5 hours, thus the EE's drift due to thermal effect stops after 6 hours. A similar but longer phenomena have been observed on a steel-made robot in [33]: robot's joint temperature stabilizes after 6 hours, thus the EE's position no longer evolved after 10 hours. To avoid any thermal drift during the measurement process, either the process should be fast enough (with regard to the time constant of the exponential decay that can model the EE's drift across time), or the robot have to be warmed up before measurement, thus almost doubling the time needed for the calibration of a robot.

In this paper, a new active learning-based method for robot calibration is proposed. The goal of this calibration method is to reduce drastically the number of data needed, while being more accurate than traditional hybrid calibration methods based on ANN. The method (depicted in Fig.1) uses the uncertainty indicator provided by the GPRs to iteratively build the training set in an active learning fashion. Thus, after an initialization step, the algorithm chooses itself the most relevant data to learn from, and when the training should stop. Since the proposed algorithm depends on several hyper-parameters (e.g. the kernel used or the stopping condition), they have been empirically investigated and finely tuned, using data from our previous work [32], for a more complete analysis and evaluation. These hyper-parameters have been used to experimentally validate the proposed method on a KUKA KR300 industrial robot for positioning error compensation, and on

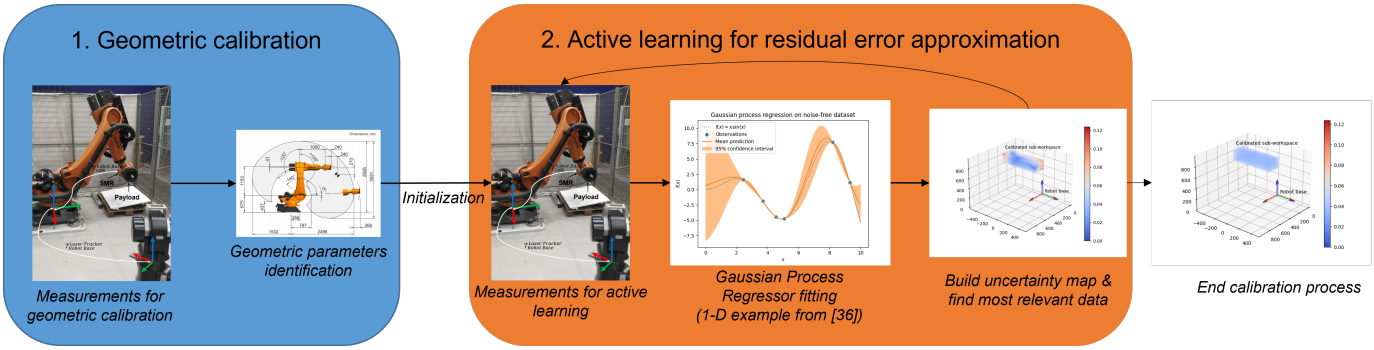


Fig. 1: Proposed method flowchart

a KUKA lbr *iiwa* R820 for both positioning and orientation accuracy enhancement. In section II, the general method is described: hybrid calibration methods are detailed, and the active learning based process for residual error approximation is presented. Section III investigates empirically the finest tuning for the method's hyper-parameters, using data from our previous work [32]. Section IV describes the experimental setup required to apply the proposed method, and shows that the proposed method is fast enough to neglect thermal drift. Experimental results are presented and discussed in section V, and section VI concludes.

II. Method description

A. Hybrid calibration

Hybrid calibration combines identification of analytical models parameters and approximation of residual errors through machine learning. The final goal is to compensate for positioning errors via an algorithm that uses an accurate mapping between the joint configuration of the robot and the position of its end-effector. In this paper, this accurate mapping is made on the first hand through forward kinematics, using identified geometric parameters, and on the other one using a GPR to predict the residual error (i.e. the difference between the measured position and the theoretical one after geometric calibration). The general method is described in Fig.2.

The geometric model used is the DH-model [34], in which four parameters (α, a, r, θ) for each joint are used to describe the transformation to the next joint. The transformation matrix between joint $i-1$ and i is:

$$T_{i-1}^{i} = \boldsymbol{Rot}(x_{i-1}, \alpha_i) \cdot \boldsymbol{Trans}(x_{i-1}, a_i) \cdot \boldsymbol{Rot}(z_i, \theta_i) \cdot \boldsymbol{Trans}(z_i, r_i) \quad (1)$$

The geometric parameters of the robot are gathered in a vector denoted ξ. The Forward Kinematics (FK) is the serial multiplication of these transformation matrices over all axes for a given joint position $\Theta = \theta_{i=1\,\dots\,n}$. Hence, for an *n*-axes robot, the forward kinematics is:

$$FK(\xi, \Theta) = T_0^n(\xi, \Theta) = \prod_{i=1}^{n} T_{i-1}^{i} \quad (2)$$

To identify the geometric parameters, the generalized Jacobian matrix (that is the matrix gathering the partial derivatives of the cartesian position with regard to the geometric parameters), denoted J, is computed. For each joint position of the set $\theta_{geometric}$, the associated theoretical cartesian position x_{th} is computed with eq. (2) and is compared to the measured cartesian position $x_{measured}$. The identified geometric parameters are:

$$\xi_{identified} = \xi_{nominal} + J \backslash \Delta x \quad (3)$$

where $\Delta x = x_{measured} - x_{th}$.

Finally, in the hybrid calibration method, the mapping between the joint position and the cartesian position of the EE is defined by the hybrid model

$$X = MB(\xi_{identified}, \Theta) + ML(\Theta) \quad (4)$$

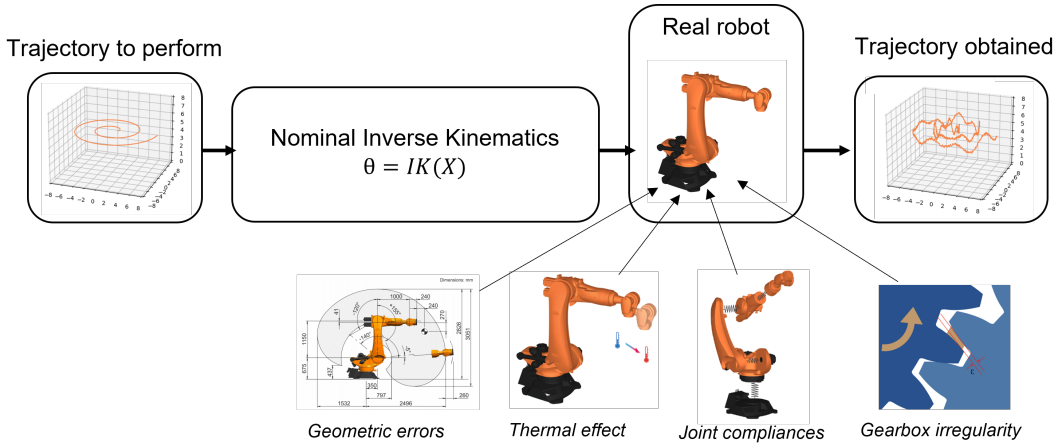
where $MB(\xi_{identified}, \Theta)$ is the *model-based* part and $ML(\Theta)$ is the *model-less* one (e.g. $ANN(\Theta)$ or $GPR(\Theta)$). In this paper, we have:

$$X = FK(\xi_{identified}, \Theta) + GPR(\Theta) \quad (5)$$

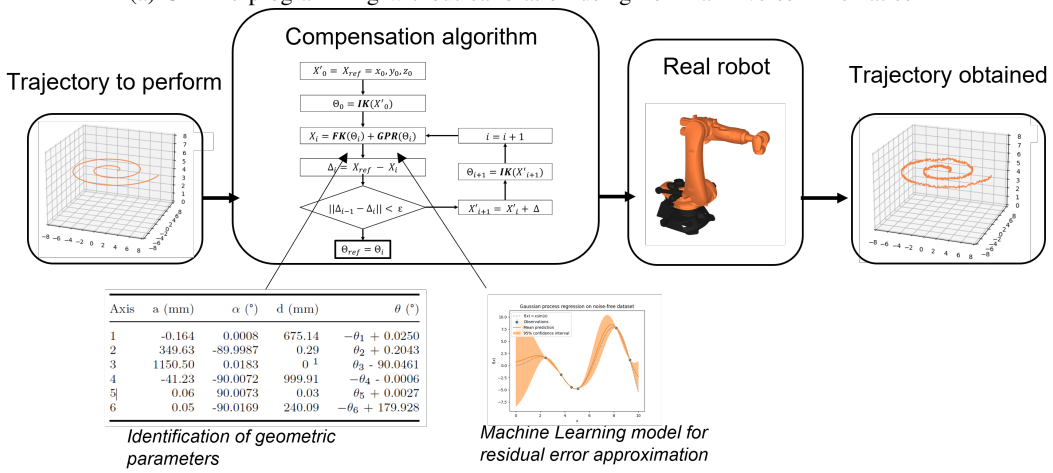
The geometrical parameters ξ must be identified first, as the GPR is to map the error between the analytical model and the measurements. This model can now be used in a compensation algorithm that computes the joint position the robot should be in to reach a desired cartesian position. The compensation algorithm is described in Fig.3. This algorithm takes as input the desired cartesian position to reach. Through the nominal Inverse Kinematics (**IK**) model, a joint position is deduced. Using the hybrid model, a theoretically reached cartesian position X_i is computed. The difference between X_{ref} and X_i (denoted Δ_i) is then deduced, and added to X_{ref} to give an intermediate cartesian position X'_i. From this cartesian position, a new joint position is given by the **IK**, and the previous steps are repeated until Δ_i no longer evolves. Finally, the last computed joint position reaches the desired cartesian position, according to the model.

B. Gaussian Process regressors

Gaussian Process Regressors are a machine learning tool used for regression by a Bayesian approach. GPR models training data as multivariate Gaussian distribution (thus a GPR is defined by its mean and covariance functions, $m(x)$ and $k(x, x')$), so that a prediction can be made on any new input via Bayesian inference [35].



(a) Off-line programming without calibration using nominal Inverse Kinematics



(b) Off-line programming with hybrid calibration using compensation algorithm

Fig. 2: Hybrid calibration principle

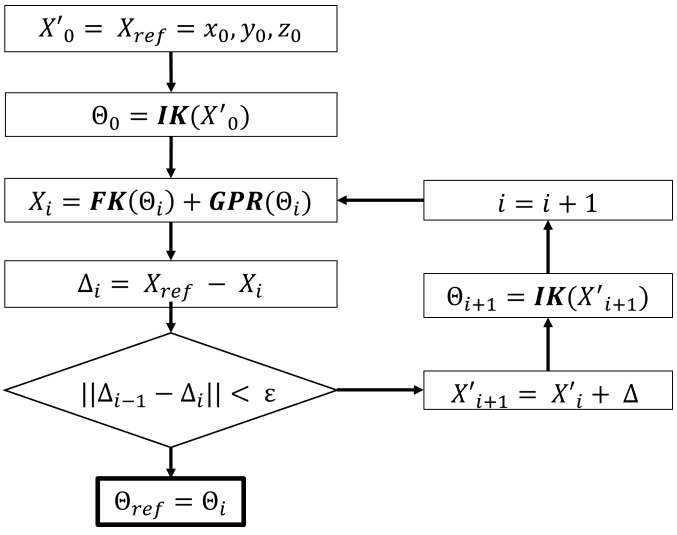


Fig. 3: Compensation algorithm

The covariance function needs to be carefully chosen. For a training set (x_t, y_t), the multivariate Gaussian distribution is:

$$y_t \sim \mathcal{N}(m(x_t), K(x_t, x_t)) \tag{6}$$

with $K(x_t, x_t)$ the covariance matrix of the inputs. Since in our case the mean function m is the difference between the measurements and the analytical model (from eq. (5)), we choose $m(x) = 0$. For any new input x_{new}, the multivariate Gaussian distribution becomes:

$$\begin{pmatrix} y_t \\ y_{new} \end{pmatrix} \sim \mathcal{N}(0, \begin{pmatrix} K(x_t, x_t) & K(x_t, x_{new}) \\ K(x_{new}, x_t) & K(x_{new}, x_{new}) \end{pmatrix}) \tag{7}$$

From Bayes' inference rule, it follows:

$$\begin{aligned} &y_{new}|y_t \sim \mathcal{N}(K(x_t, x_{new})K(x_t, x_t)^{-1}y_t, \\ &K(x_{new}, x_{new}) - K(x_t, x_{new})K(x_t, x_t)^{-1}K(x_{new}, x_t)) \end{aligned} \tag{8}$$

From eq. (8), the mean and the standard deviation for any new input can be deduced from observations. The mean is the predicted value, and the standard deviation can be interpreted as an uncertainty indicator on the prediction: the lower the standard deviation is, the more confident is the prediction. This uncertainty can be used in an active learning based algorithm to reduce the number of observation data needed, as presented in [31]. The method proposed in this paper have been developed in Python, using the *scikit-learn* library [36] for the GPR's design and training.

C. Active learning based algorithm

Active learning is a sub-part of machine learning where the model is able to actively choose the most relevant data to learn from, rather than being passively trained on a pre-selected dataset. In this context, active learning aims at training a model with the smallest amount of labeled data while maintaining a high level of performance. Starting with very few data coming from measurements in a sparse meshgrid of the workspace, the uncertainty map provided by a GPR approach is exploited to build an autonomous iterative active learning based process. Hence only relevant new data are acquired, until the stopping condition is reached.

The process can be separated into two parts: an initialization phase and an iterative phase. The initialization phase consists first in griding the sub-workspace that has to be calibrated. The finest the grid is, the more accurate will be the choice of new relevant observations. Then, a few samples have to be selected among the grid (denoted $\mathcal{S}$) to form the initial training set. After the training of a GPR over this training set, the GPR builds an uncertainty map (through a prediction over $\mathcal{S}$), and choose the next data to acquire as the one with the highest standard deviation (i.e. the highest uncertainty). This last step is repeated until the stopping condition is met. This process is summarized in Algorithm 1. The overall proposed method is illustrated in Fig.1

Algorithm 1 Active learning based process

Require: $\mathcal{S}$, $\Theta_{measured}$, $X_{measured}$
  $fit\, GPR(\Theta_{measured},\ X_{measured})$
  repeat
    $m,\ \sigma \leftarrow GPR(\mathcal{S})$
    $\theta_{next} \leftarrow \mathcal{S}(max(\sigma))$
    $X_{next} \leftarrow measurement(\theta_{next})$
    $\Theta_{measured},\ X_{measured} \leftarrow append(\theta_{next}, X_{next})$
    $fit\, GPR(\Theta_{measured},\ X_{measured})$
  until $stoppingcondition$

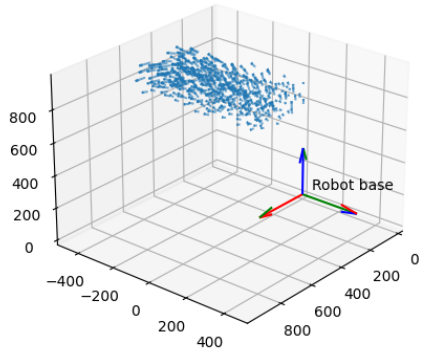


Fig. 4: Error vector field across the calibrated sub-workspace. For displaying purposes, all the errors have been amplified by the same factor.

III. Method's hyper-parameters tuning

In the proposed algorithm, several hyper-parameters must be carefully tuned. First, as stated in section II-B, a GPR is defined by its mean function, and its covariance function also called kernel. GPR's kernels can be the sum of well-known kernels, usually a regression kernel (such as the Matérn kernel or the Radial Basis Function) and a noise kernel, that is to model the measurement noise. Second, as the calibrated sub-workspace may be large, the initial observation data distribution may be of importance. The stopping condition is also to be considered. In the context of hybrid calibration, the influence of the geometric calibration made in first instance of the overall hybrid calibration process will be studied as well. All these elements will be investigated using measurements made for our previous work [32]. This dataset is made up of 2000 measurement points, evenly spaced in a sub-workspace of the KUKA lbr *iiwa*. The discretized sub-workspace (Θ, X) discussed before is then fully known, so the algorithm can be run offline for parameters tuning purposes. Finally, the contribution of the active learning based algorithm using GPR is showed through comparison with other *model-less* choices.

A. Regression kernel used

As stated previously, GPR's kernel are most of the time the sum of some usual kernels: one for regression purposes, the other one for measurement noise modeling. Here, some commonly-used regression kernels will be compared. Based on observations made on an interactive plot (depicted in Fig.4), the error vector field that is to be approximated by the GPR looks smooth and differentiable. For such cases, the *Matérn* covariance function family are known to be effective. It is defined as:

$$k_\nu(x,y) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\rho} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}r}{\rho} \right) \tag{9}$$

with ρ the range parameter, ν the smoothness parameter, Γ the gamma function, and K_ν the modified Bessel function of the second kind of order ν. The design parameter ν represents the differentiability of the function to approximate: $\nu = 1.5$ corresponds to a function once differentiable, $\nu = 2.5$ a function twice differentiable, and so on. The special case $\nu \to \infty$ is the well-known Radial Basis Function (RBF).

These three kernels ($Matérn(\nu = 1.5)$, $Matérn(\nu = 2.5)$, RBF) have been compared on the test set and compared. To do so, 100 points have been extracted randomly from the dataset to form the test set. Then, another 5 points have been extracted randomly to initialize the algorithm. Finally, the algorithm have been ran over 45 iterations, so that the training set is finally composed of 50 points. For this comparison, no stopping conditions have been used, to fully investigate the performances of each one of the kernels. Finally, the accuracy of the model (eq. (5)) is compared over the test set to the measured cartesian position for the corresponding joint positions. The results are summed up on Fig.5. The performances are not very sensitive to the kernel choice

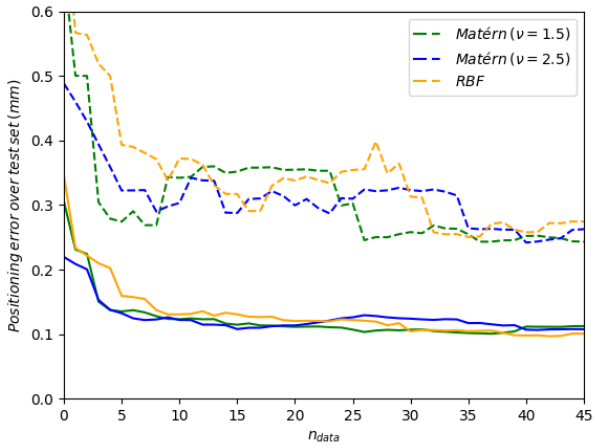


Fig. 5: Comparison of the different kernel for positioning error approximation. The full lines represent the evolution of the mean error over the test set along the iterations of the algorithm, the dotted lines represent the evolution of the max error.

between the three studied. As the initialization is randomized, this test have been ran several time, and the results are always consistent with the one presented. We decided to choose the $Matérn(\nu = 1.5)$ kernel in the method.

B. Noise kernel influence

Since the observation data used for training the GPR come from real measurements, they come with a measurement noise. Assuming this noise is Gaussian, with a null mean and a variance σ^2, the kernel used in the GPR could be added to a noise kernel that is to take into account this measurement noise. By this way, our model will not fit exactly on the observation data, but should accept a slight residue to enhance the global accuracy. However, this noise level is not exactly known, since the accuracy of the measurement device depends on the distance between the Laser Tracker and the Spherical Mounted Reflector (SMR), thus is not constant, and on the repeatability of the robot (which depends itself as well on the accuracy of the measurement device). Consequently, 5 different scenarios have been considered, to study the influence of the value σ for the noise kernel's settings on the proposed calibration process performances:

- $\sigma = 0$, to assess the need of a noise kernel
- $\sigma = 0.04$, corresponding to the sum of the variance of the measurement device, given by the acquisition software, and the variance of the measured points during a repeatability test, carried out on the same robot in the sub-workspace calibrated. This can be considered as a *well-evaluated* noise level, with regard to the available data and the different tests carried out
- $\sigma = 0.01$, corresponding to an *under-evaluated* noise level
- $\sigma = 0.08$, corresponding to an *over-evaluated* noise level
- $adaptative\ \sigma$, corresponding to letting the GPR find the optimal value of σ during the training phases

The algorithm have been tested following the same condition than the previous test. According to section III-A, the regression kernel used here is $Matérn(\nu = 1.5)$. The result of this study is depicted on Fig.6. One can note that an over-evaluated noise does not benefit the algorithm: the mean positioning error is higher than all other scenarios, due to over-residues. Since there is no big difference between the other settings, adaptative noise settings is prefered, as it requires no *a priori* knowledge on the noise level of the measurement setup.

C. Initialization

The initial data distribution across $\mathcal{S}$ is now investigated. As the objective is to get only the most relevant data for the training phase, the size of the initial dataset should be as low as possible. In the following, we set the initial size to 5, as an empiric compromise between convergence speed and number of data required. Since this initial size is very small, choosing the points randomly could lead to different results from one initialization to another. Instead of choosing randomly, one could use low-discrepancy series, so that the data are deterministicaly chosen and more evenly spaced in the sub-workspace.

These two different methods of initialization are compared, with a similar experiment than before. The kernel used for the GPR is the $Matérn(\nu = 1.5)$ one, added to a noise kernel with adaptative noise level. The evolution mean and max error over the test set across the iterations of the algorithm is depicted in Fig.7. One can see that Halton sequence performs better than random initialization. Moreover, the experiment have been carried out several times, since the random initialization could provide different results, but the Halton sequence always performed better, and is more consistent.

D. Influence of geometric calibration

Since the objective of the proposed method is being more data-efficient, one could think that geometric calibration would require more data than it would save for GPR training. Thus, two models are proposed, both based on eq. (5): the first one uses $\xi_{nominal}$ (this corresponds to *model-less* calibration), and the other one uses $\xi_{identified}$. The same experiment than before have been carried out, using $Matérn(\nu = 1.5)$ kernel added to a noise kernel with adaptative noise level, and initialized with 5 data chosen from low discrepancy series. The mean and max error evolution over the test set across the iterations of the active learning based algorithm is depicted in Fig.8. One can see on Fig.8 that using identified geometric parameters instead of nominal ones lead to a faster convergence. As usually several sub-workspaces can be calibrated for several sub-tasks, it is overall a better compromise to have geometric calibration performed before GPR training.

E. Stopping condition

Since all other parameters of the proposed algorithm have now been tuned, the stopping condition can now be investigated. The objective is to find an intrinsic value that would

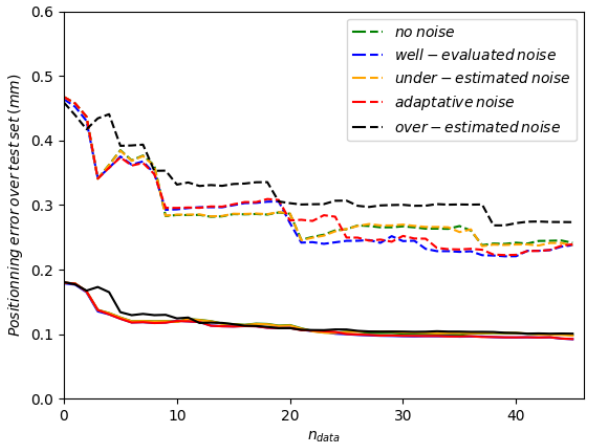


Fig. 6: Comparison of different noise kernel's settings. The full lines represent the evolution of the mean error over the test set along the iterations of the algorithm, the dotted lines represent the evolution of the max error.

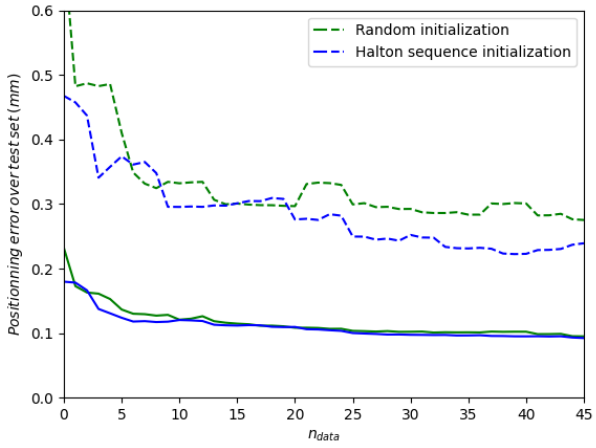


Fig. 7: Comparison of the different initialization methods. The full lines represent the evolution of the mean error over the test set along the iterations of the algorithm, the dotted lines represent the evolution of the max error.

follow the evolution of the mean error over the test set, so that during real experiments there is no need to have additional measurements to monitor the evolution of the algorithm's performances across iterations. Indeed, on all previous experiments, one can see that the mean error reaches quickly a plateau, meaning that the real number of data required is low compared to the one fixed for the experiment needs. Two indicators are proposed and investigated. The first one is the evolution of the maximal uncertainty over our uncertainty map, which corresponds to a threshold value above whom the self-confidence of the GPR is judged insufficient, so additional data is needed. The second one is the standard deviation of the uncertainty (uncertainty spread), so that the algorithm iterates until the uncertainty is homogeneous over the whole calibrated

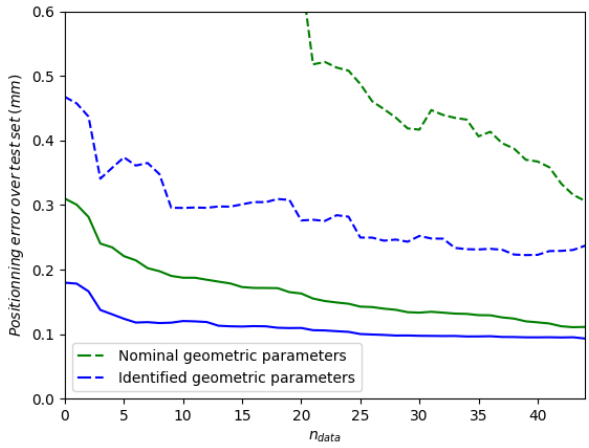


Fig. 8: Comparison of *model-less* and *hybrid* calibration. The full lines represent the evolution of the mean error over the test set along the iterations of the algorithm, the dotted lines represent the evolution of the max error.

sub-workspace. Fig.9 shows the evolution of the mean error, the maximal uncertainty and the uncertainty spread across iterations. On this figure, one can clearly see that the mean error and the uncertainty spread seems highly correlated. As a stopping condition, one can iterate until the uncertainty spread no longer evolves.

F. Performances

Since every parameters have been tuned, the overall performances of the proposed method can be establish on the test set. Thus, according to previous sections, our GPR is made up of a *Matérn* kernel with $\nu = 1.5$, a noise kernel with an adaptative noise level. Initial data are selected using Halton set, a low discrepancy serie. The algorithm stops itself when the standard deviation of the uncertainty over $\mathcal{S}$ no longer evolves, i.e. the uncertainty is homogeneous. Using these parameters, the proposed method needed 21 iterations, leading to a final training set of 26 data. Fig.10 shows the evolution of the standard deviation of the uncertainty over $\mathcal{S}$. On Fig.11, one can see that the uncertainty is homogeneous at the end of the method.

To assess the effectiveness of the proposed method, its *model-less* part is modified for comparison purposes:

- An ANN trained with the exact same data after the active learning phase of a GPR
- An ANN trained with 400 data chosen randomly, which is a good compromise between accuracy and dataset-size according to [32] (since experiments are carried on the exact same robot in a similar sub-workspace)
- A GPR trained with the same number of data than after the active learning phase of our method but chosen randomly instead of according to the data relevance, thus highlighting the benefits of the active learning method

The results are summarized in Tab.I. The proposed method performs better than every other method it was compared to.

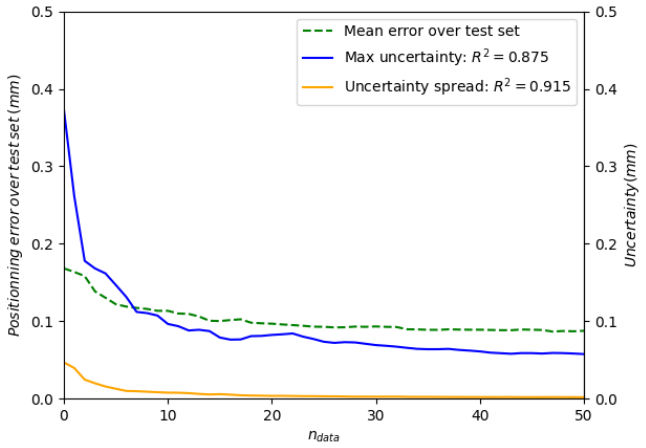


Fig. 9: Comparison of two stopping conditions that are expected to fit the mean error over the test set. Pearson's correlation coefficient is in legend as key indicator

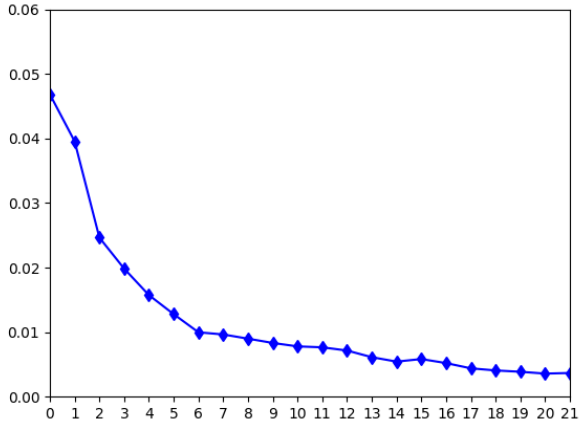
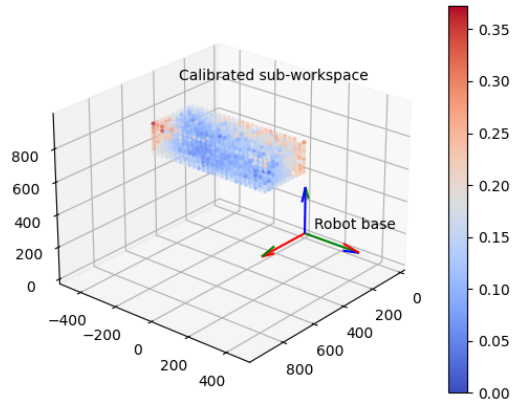


Fig. 10: Evolution of the standard deviation of the uncertainty indicator using data from [32].

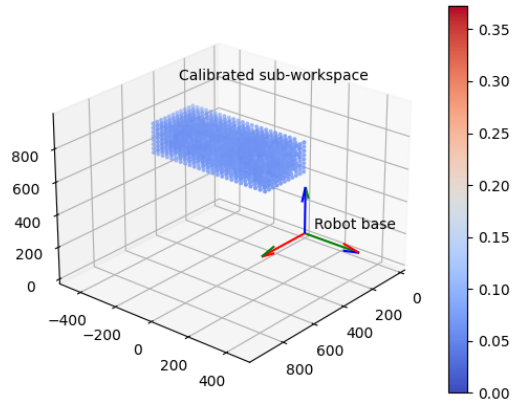
One can see that GPR trained with random data provides better results than ANN trained with the same number of data. However, both methods are less accurate than the one proposed. Moreover, the proposed method provides better results than training an ANN with 15 times more data.

IV. Experimental setup

To assess the effectiveness of the proposed method in real industrial scenarios, it must be tested on various robots using the compensation algorithm described in section II. First, the method will be applied on a traditional industrial robot, a KUKA KR300, for positioning accuracy enhancement. Second, it will be tested on a collaborative robot, the KUKA *iiwa* (the same one that provided data in the previous section), for both positioning and orientation error compensation. For both robot, the robot base frame have been identified using the same method than in [37], and all measurement have been made using a Laser Tracker API III, which accuracy is $15\mu m + 5\mu m/m$.



(a) Uncertainty map after initialization.



(b) Uncertainty map at the end of the process.

Fig. 11: Evolution of uncertainty over the calibrated sub-workspace using dataset from [32].

TABLE I: Comparison between GPR and ANN

Method	Mean error (mm)	Max error (mm)
Proposed method (26 training data)	0.1003	0.2079
ANN (using same data)	0.2267	0.5098
ANN (400 data selected randomly)	0.1091	0.2242
GPR (26 data selected randomly)	0.1767	0.3818

A. KUKA KR300

The experiments have been set up according to section II. First, geometric calibration of the robot have been performed, using measurements made across the whole workspace of the robot. The nominal and identified geometric parameters of the KR300 are described in Tab.II and Tab.III. Then, the GPR have been designed according to the previous section: it is made up of a Matérn kernel with $\nu = 1.5$ added to a noise level kernel

TABLE II: DH model of the KR300 robot (nominal geometric parameters)

Joint	a (mm)	α (°)	r (mm)	θ (°)
1	0	0	675	$-\theta_1$
2	350	-90	0	θ_2
3	1150	0	0	θ_3 - 90
4	-41	-90	1000	$-\theta_4$
5	0	90	0	θ_5
6	0	-90	240	$-\theta_6$ + 180

TABLE III: DH Model of the KR300 robot (identified geometric parameters)

Joint	a (mm)	α (°)	r (mm)	θ (°)
1	-0.164	0.0008	675.14	$-\theta_1$ + 0.0250
2	349.63	-89.9987	0.29	θ_2 + 0.2043
3	1150.50	0.0183	0 *	θ_3 - 90.0461
4	-41.23	-90.0072	999.91	$-\theta_4$ - 0.0006
5	0.06	90.0073	0.03	θ_5 + 0.0027
6	0.05	-90.0169	240.09	$-\theta_6$ + 179.928

*This parameter cannot be identified because of a loss of rank in the generalized jacobian, as explained in [38]

with adaptative noise. Since the calibrated sub-workspace is significantly bigger than the one of the previous section, 10 initial data have been selected from a Halton sequence. The general setup is depicted in Fig.12.

B. KUKA lbr iiwa R820

The experimental setup for the iiwa is similar to the one of the KR300. Geometric calibration has been performed as well in first instance, using measurements made across the whole workspace. The nominal and identified geometric parameters of the iiwa are gathered in Tab.IV and Tab.V. The GPR design is the same than for the KR300, with the exception that it outputs both positioning and orientation error estimation. The orientation is measured using 3 SMR whose positions are fully known in the EE's frame, since the Laser Tracker used can not naturally measure orientation. As the 3 SMR positions are fully known, for any new joint configuration, one simply needs to measure their new positions, and fit them to their reference one in the EE's frame to reconstruct it, as done in [25]. Thus, the tranformation matrix between robot's base and the EE's frame is known, and orientation can be compared as well as position. Initial data have been selected similarly than previously, using 5 samples from a Halton sequences. The general setup used for the experiments is depicted in Fig.13, and, more specifically, the orientation measurement process is depicted in Fig.14

C. Thermal drift

Since measurement processes for *model-less* and *hybrid* calibration can be time-consuming (because they are data-consuming), thermal drift can occur. As the proposed method is data-efficient, the measurement process may be fast enough to avoid any thermal drift. In [13], [33], both joint's temperature and drift of the robot's EE are modeled by an exponential decay, of the form $T(t) = A_T e^{\frac{-t}{\tau_T}} + T_0$ and $X(t) = A_X e^{\frac{-t}{\tau_X}} + X_0$, with different time constants. Both

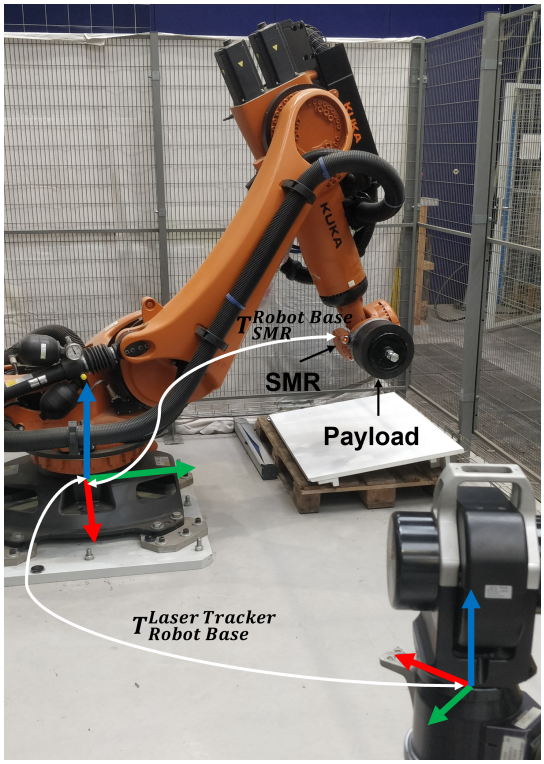


Fig. 12: Experimental setup for the KR300

TABLE IV: DH model of the *iiwa* (nominal geometric parameters)

Joint	a (mm)	α (°)	r (mm)	θ (°)
1	0	0	360	θ_1
2	0	-90	0	θ_2
3	0	90	420	θ_3
4	0	90	0	θ_4
5	0	-90	400	θ_5
6	0	-90	0	θ_6
7	0	90	152	θ_7

TABLE V: DH model of KUKA LBR *iiwa* 14R820 (identified geometric parameters)

joint	a (mm)	α (deg)	r (mm)	θ (deg)
1	-0.199	0.004	358.687	θ_1 + 0*
2	-0.265	-90.005	0.109	θ_2 + 0.032
3	-0.012	90.031	420.498	θ_3 - 0.027
4	0.004	89.976	-0.292	θ_4 + 0.006
5	0.146	-89.982	400.417	θ_5 + 0.025
6	0.015	-90.013	-0.066	θ_6 + 0.007
7	0 *	90*	152*	θ_7 + 0 *

*This parameter cannot be identified because of a loss of rank in the generalized jacobian, as explained in [38]

time constant have been measured on both robots, thanks to the Laser Tracker and internal temperature sensors, and are summed up in Tab.VI. The amplitude of the thermal drift does not exceed 0.3mm for both robot. Since the measurement process for applying the method were $4min$ for the KR300 and $7min$ for the *iiwa*, and with regard to the time constant of the EE's evolution across time, the thermal drift during the measurement process does not exceed $30\mu m$, which is neglectable.

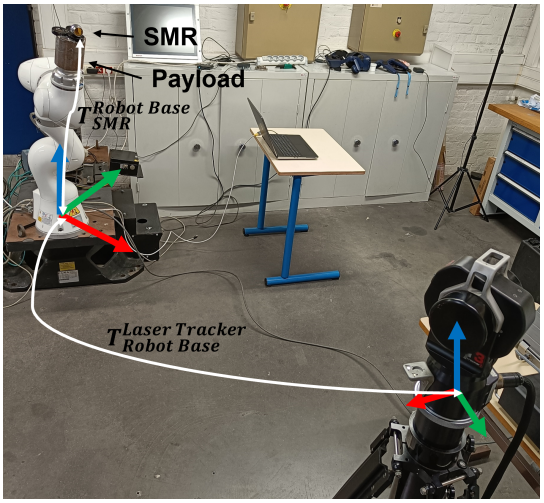


Fig. 13: Experimental setup used for the *iiwa*.

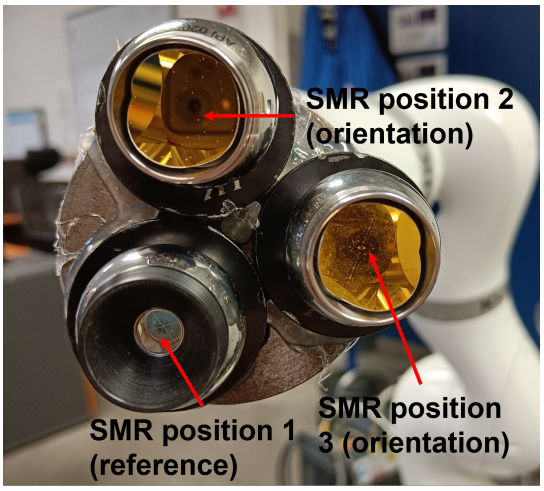


Fig. 14: Full pose measurement setup. SMR position 1 is used as pose center, SMR position 2 and 3 are used as reference to build the full pose.

TABLE VI: Time constants of the temperature and the EE's position evolution across time.

	KR300	iiwa
τ_T (h)	1.66	1.04
τ_X (h)	2.91	1.17

V. Results

A. KUKA KR300

With the previously-described calibration process, 24 measurements were used to train the GPR to compensate for residual errors after geometric calibration on the KR300. Indeed, after the initialization phase (here 10 points), one can see on Fig.15 that the uncertainty's standard deviation stopped its evolution after the 14^{th} iteration. On Fig.16, the uncertainty at the end of the process is well homogeneous on the whole sub-workspace, while being heterogeneous after initialization. However, looking at Fig.15, variation of the stopping condition could be low even during the convergence, so that early

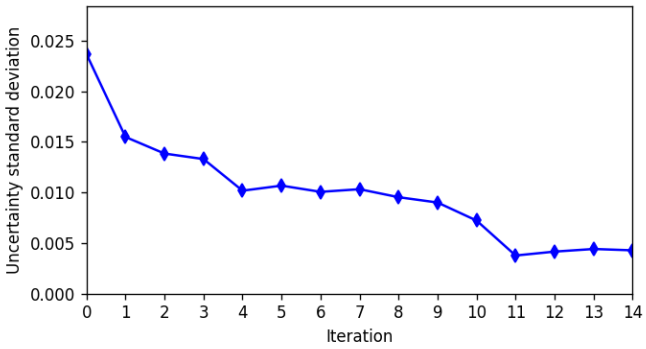


Fig. 15: Evolution of the standard deviation of the uncertainty along the experimental protocol for the KR300.

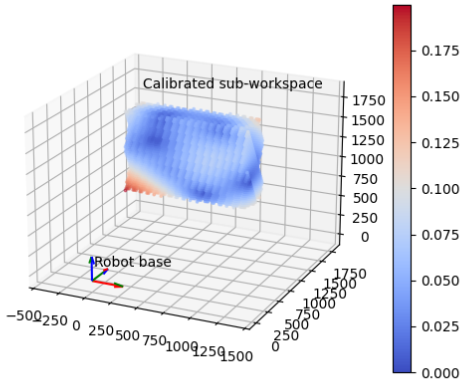
TABLE VII: Performances of the proposed method compared to no calibration and geometric calibration on the KR300.

	Mean (mm)	Max (mm)
Error between nominal model and measured position on training set	3.0603	5.2859
Error between identified geometric model and measured position on training set	0.4607	0.8012
Error after proposed method	0.0803	0.1819

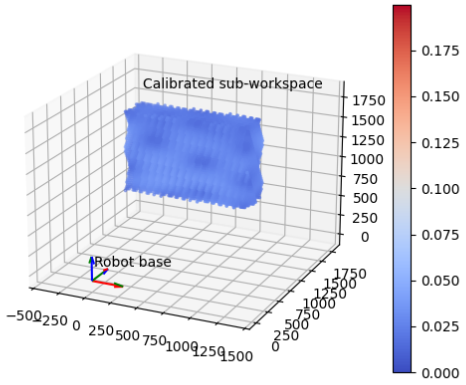
stopping could unfortunately happen. It may be safer to extend the stopping condition on a larger window, though ending the process on few non-relevant points. To assess the effectiveness of the proposed method, 30 cartesian positions were randomly generated, and their corresponding joint positions were computed using the compensation algorithm depicted in Fig.3. One can see on Tab.VII and Fig.17 that the proposed method can reduce the mean positioning error by 98%.

B. KUKA lbr iiwa R820

Following the experimental process described in the previous section, the *iiwa* were calibrated using only 19 measurements. Fig.18 shows the evolution of the stopping conditions during the experiments. As the standard deviation of the uncertainty over the workspace stopped evolving, the experiments were stopped at the 14th iterations. Hence, the total training set is made up of 19 measurements. One can see on Fig.19 that the uncertainty has evolved across the iterations, and is homogeneous at the end of the process. Similarly than for the KR300, 30 cartesian positions were randomly generated, and their corresponding joint positions were computed using the compensation algorithm depicted in Fig.3. Comparing the measured reached cartesian position and the desired one, the accuracy of the protocol is summed up in Tab.VIII. For comparison purposes, Tab.IX shows the positioning and orientation error over the training set, after geometric calibration. Even though these values have been computed over only 19 points, the trend is clear: the positioning error have been reduced by 90% after our protocol, while the orientation error have been reduced by 92% to 98% (depending on the axis). Moreover, one can see on Fig.20 and Fig.21 that the error distribution is much more tighter after applying the proposed method than before.



(a) Uncertainty map after initialization



(b) Uncertainty map at the end of the protocol

Fig. 16: Evolution of uncertainty over the calibrated sub-workspace for the KR300

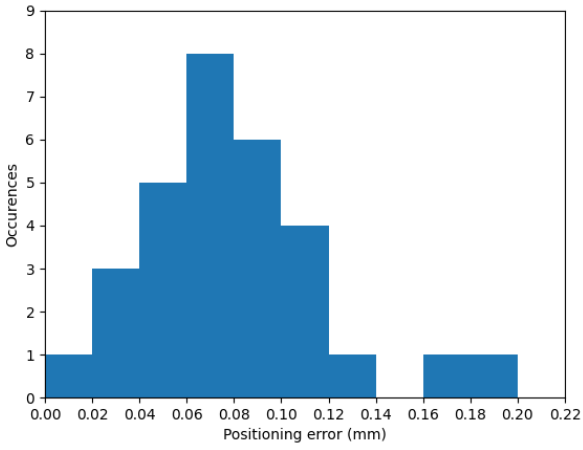


Fig. 17: Error distribution after calibration, over 30 validation points, using the compensation algorithm on the KR300.

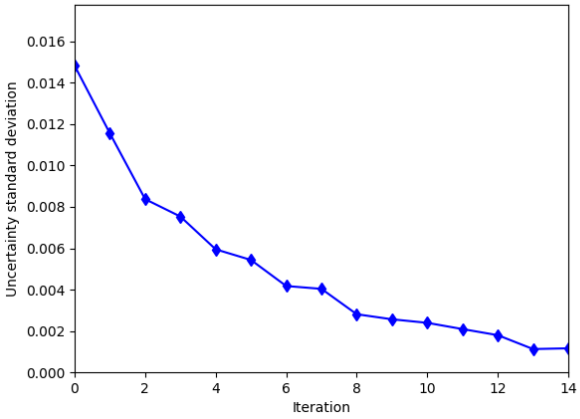
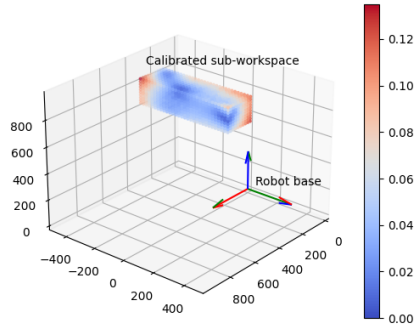
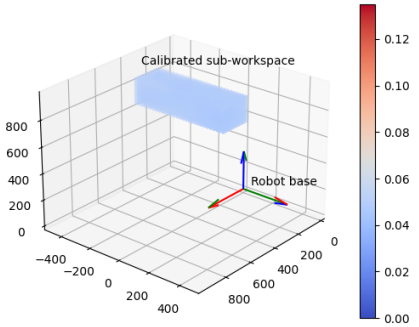


Fig. 18: Evolution of the standard deviation of the uncertainty along the experimental protocol for the *iiwa*



(a) Uncertainty map after initialization



(b) Uncertainty map at the end of the protocol

Fig. 19: Evolution of uncertainty over the calibrated sub-workspace for the *iiwa*

VI. Conclusion

In this paper, a data-efficient, accurate and fast active learning based method for robotic hybrid calibration is proposed. It relies on one hand on the identification of the geometric parameters and on the other hand on a Gaussian Process Regressor that is to approximate the residual errors, trained in

TABLE VIII: Positioning and orientation error after hybrid calibration on the *iiwa*.

	Mean	Max
Positioning error	0.109 mm	0.266 mm
Orientation error around $\vec{x}$	0.006°	0.017°
Orientation error around $\vec{y}$	0.006°	0.022°
Orientation error around $\vec{z}$	0.007°	0.020°

TABLE IX: Positioning and orientation error over training set (after geometric calibration) on the *iiwa*.

	Mean	Max
Positioning error	0.644 mm	0.98 mm
Orientation error around $\vec{x}$	0.036°	0.089°
Orientation error around $\vec{y}$	0.100°	0.185°
Orientation error around $\vec{z}$	0.026°	0.063°

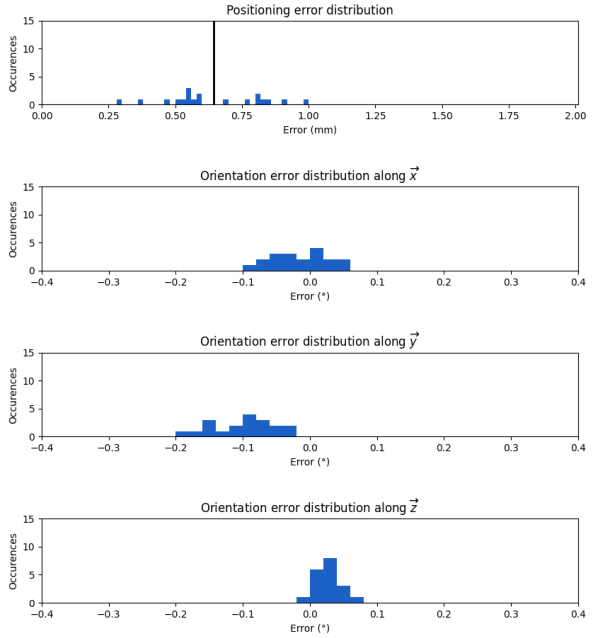


Fig. 20: Error distribution before calibration, over the training set of 19 points, on the iiwa

an active learning fashion. From a very few initialization data that are used to train a first version of the GPR, an uncertainty map can be built, based on the GPR prediction over the sub-workspace that is to be calibrated. From this uncertainty map, the most relevant new data can be selected autonomously, measured and added to the training set, which is complete when the uncertainty map becomes homogeneous. Experimental validation using a compensation algorithm shows that this method is able to reduce the positioning error on a traditional industrial robot, the KR300, by 97% using only 24 training data. The mean value of the error after calibration has been reduced to 0.08mm. Similarly, the positioning and the orientation error of a collaborative robot, the KUKA lbr *iiwa* R820, is reduced by 90%, using only 19 data for the GPR's training. Since the measurement process is fast, no thermal effects bias the training data. Thus, compared to state-of-the-art method on hybrid calibration, the proposed method offers a better accuracy.

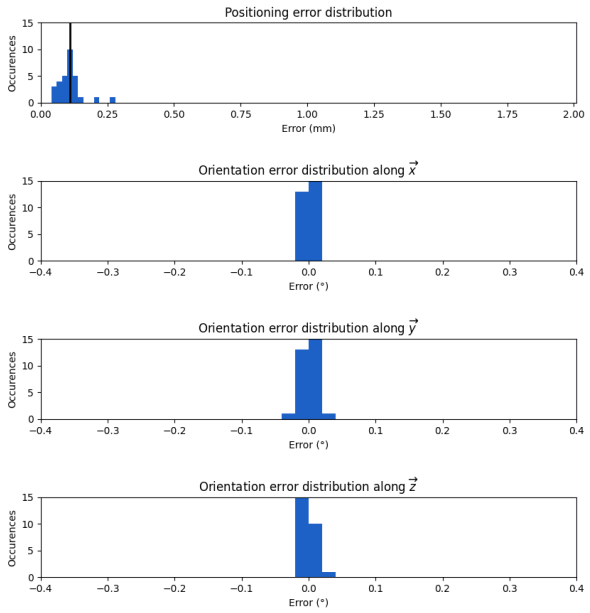


Fig. 21: Error distribution after calibration, over 30 validation points, on the iiwa

REFERENCES

[1] A.Y. Elatta, L.P. Gen, F.L. Zhi, Y. Daoyuan, and L. Fei. "An overview of robot calibration". In: *Information Technology Journal* 3 (Jan. 2004), pp. 74–78.

[2] B. Mooring, Zvi Roth, and M. Driels. "Fundamentals of Manipulator Calibration". In: (Jan. 1991), p. 27.

[3] Jia-Qing Xuan, Sun-Han Xu, et al. "Review on kinematics calibration technology of serial robots". In: *International journal of precision engineering and manufacturing* 15.8 (2014), pp. 1759–1774.

[4] Zhibin Li, Shuai Li, and Xin Luo. "An overview of calibration technology of industrial robots". In: *IEEE/CAA Journal of Automatica Sinica* 8.1 (2021), pp. 23–36.

[5] Yu Sun and John M Hollerbach. "Observability index selection for robot calibration". In: *2008 IEEE international conference on robotics and automation*. IEEE. 2008, pp. 831–836.

[6] Ahmed Joubair and Ilian A Bonev. "Comparison of the efficiency of five observability indices for robot calibration". In: *Mechanism and Machine Theory* 70 (2013), pp. 254–265.

[7] J. Kenneth Salisbury. "Active stiffness control of a manipulator in cartesian coordinates". In: *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes* (1980), pp. 95–100.

[8] Wisama Khalil and Sébastien Besnard. "Geometric Calibration of Robots with Flexible Joints and Links". In: *Journal of Intelligent and Robotic Systems* 34 (Aug. 2002), pp. 357–379.

[9] Claire Dumas, Stéphane Caro, Sébastien Garnier, and Benoît Furet. "Joint Stiffness Identification of Six-revolute Industrial Serial Robots". In: *Robotics and Computer-Integrated Manufacturing* 27.4 (Aug. 2011), pp. 881–888.

[10] Adel Olabi, Mohamed Damak, Richard Bearee, Olivier Gibaru, and Stephane Leleu. "Improving the accuracy of industrial robots by offline compensation of joints errors". In: *2012 IEEE International Conference on Industrial Technology*. 2012, pp. 492–497.

[11] Pierre Besset, Adel Olabi, and Olivier Gibaru. "Advanced calibration applied to a collaborative robot". In: *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*. 2016, pp. 662–667.

[12] Timothy Tuttle. "Understanding and Modeling the Behavior of a Harmonic Drive Gear Transmission". In: 1992.

[13] Nenad Kircanski, Andrew A. Goldenberg, and S. Jia. "An Experimental Study of Nonlinear Stiffness, Hysteresis, and Friction Effects in Robot Joints with Harmonic Drives and Torque Sensors". In: vol. 16. Jan. 1993, pp. 326–340.

[14] Alexandr Klimchik, Yier Wu, Anatol Pashkevich, Stéphane Caro, and Benoît Furet. "Optimal Selection of Measurement Configurations for Stiffness Model Calibration of Anthropomorphic Manipulators". In: *Applied Mechanics and Materials* 162 (Mar. 2012), pp. 161–170.

[15] Alexandr Klimchik, Stéphane Caro, and Anatol Pashkevich. "Optimal pose selection for calibration of planar anthropomorphic manipulators". In: *Precision Engineering* 40 (2015), pp. 214–229.

[16] Kenan Deng, Dong Gao, Shoudong Ma, Chang Zhao, and Yong Lu. "Elasto-geometrical error and gravity model calibration of an industrial robot using the same optimized configuration set". In: *Robotics and Computer-Integrated Manufacturing* 83 (2023), p. 102558. ISSN: 0736-5845. DOI: https : / / doi . org / 10 . 1016 / j . rcim . 2023 . 102558. URL: https : / / www . sciencedirect . com / science / article / pii / S0736584523000340.

[17] Praveen Kumar et al. "Artificial neural network based geometric error correction model for enhancing positioning accuracy of a robotic sewing manipulator". In: *Procedia Computer Science* 133 (2018), pp. 1048–1055.

[18] N. Takanashi. "6 DOF manipulators absolute positioning accuracy improvement using a neural-network". In: *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. 1990, 635–640 vol.2.

[19] Josin, Charney, and White. "Robot control using neural networks". In: *IEEE 1988 International Conference on Neural Networks*. 1988, 625–631 vol.2.

[20] Guanbin Gao, Hongwei Zhang, Hongjun San, Xing Wu, and Wen Wang. "Modeling and error compensation of robotic articulated arm coordinate measuring machines using BP neural network". In: *Complexity* 2017 (2017).

[21] Seiji Aoyagi, Atsushi Kohama, Yasutaka Nakata, Yuki Hayano, and Masato Suzuki. "Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm-". In: (2010), pp. 5660–5665.

[22] Hoai-Nhan Nguyen, Jian Zhou, and Hee-Jun Kang. "A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network". In: *Neurocomputing* 151 (2015), pp. 996–1005. ISSN: 0925-2312.

[23] Hoai-Nhan Nguyen, Phu Nguyen Le, and Hee-Jun Kang. "A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique". In: *Advances in Mechanical Engineering* 11 (Jan. 2019), p. 168781401882293.

[24] Gang Zhao, Pengfei Zhang, Guocai Ma, and Wenlei Xiao. "System identification of the nonlinear residual errors of an industrial robot using massive measurements". In: *Robotics and Computer-Integrated Manufacturing* 59 (2019), pp. 104–114. ISSN: 0736-5845.

[25] Stefan Gadringer, Hubert Gattringer, Andreas Müller, and Ronald Naderer. "Robot Calibration combining Kinematic Model and Neural Network for enhanced Positioning and Orientation Accuracy". In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 8432–8437. ISSN: 2405-8963.

[26] Jen-Chung Hsiao, Kumar Shivam, I-Fang Lu, and Tai-Yan Kam. "Positioning accuracy improvement of industrial robots considering configuration and payload effects via a hybrid calibration approach". In: *IEEE Access* 8 (2020), pp. 228992–229005.

[27] Ha Xuan Nguyen, Hung Quang Cao, Ty Trung Nguyen, Thuong Ngoc-Cong Tran, Hoang Ngoc Tran, and Jae Wook Jeon. "Improving Robot Precision Positioning Using a Neural Network Based on Levenberg Marquardt–APSO Algorithm". In: *IEEE Access* 9 (2021), pp. 75415–75425. DOI: 10 . 1109 / ACCESS . 2021 . 3082534.

[28] Wei Jing, Pey Yuen Tao, Guilin Yang, and Kenji Shimada. "Calibration of industry robots with consideration of loading effects using Product-Of-Exponential (POE) and Gaussian Process (GP)". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 4380–4385. DOI: 10.1109/ICRA.2016.7487636.

[29] Raffaello Camoriano, Silvio Traversaro, Lorenzo Rosasco, Giorgio Metta, and Francesco Nori. "Incremental semiparametric inverse dynamics learning". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 544–550.

[30] Diego Romeres, Mattia Zorzi, Raffaello Camoriano, and Alessandro Chiuso. "Online semi-parametric learning for inverse dynamics modeling". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 2945–2950.

[31] Edoardo Pasolli and Farid Melgani. "Gaussian process regression within an active learning scheme". In: *2011 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2011, pp. 3574–3577.

[32] Maxime Selingue, Adel Olabi, Stéphane Thiery, and Richard Béarée. “Experimental Analysis of Robot Hybrid Calibration Based on Geometrical Identification and Artificial Neural Network”. In: *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*. 2022.

[33] Adrien Le Reun, Kévin Subrin, Anthony Dubois, and Sébastien Garnier. “Thermal drift and backlash issues for industrial robots positioning performance”. In: *Robotica* 40.9 (2022), pp. 2933–2952. DOI: 10.1017/S0263574721002022.

[34] J. Denavit and R.S. Hartenberg. “Notation for lower-pair mechanisms based on matrices”. In: *A. Kinematic, ASME Journal of Applied Mechanics* 22 (Jan. 1995), pp. 215–221.

[35] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[37] Yuanfan Zeng, Wei Tian, Dawei Li, Xiaoxu He, and Wenhe Liao. “An error-similarity-based robot positional accuracy improvement method for a robotic drilling and riveting system”. In: *The International Journal of Advanced Manufacturing Technology* 88.9 (Feb. 2017), pp. 2745–2755. ISSN: 1433-3015. DOI: 10.1007/s00170-016-8975-8. URL: https://doi.org/10.1007/s00170-016-8975-8.

[38] Etienne Dombre and Wisama Khalil. *Robot Manipulators: Modeling, Performance Analysis and Control*. Ed. by London ISTE Ltd. Control Systems, Robotics and Manufacturing Series. 2007, p. 448.