2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Piecewise-weighted RANSAC Method Utilizing Abandoned Hypothesis Model Information with a New Application on Robot Self-calibration

Jianhui He[1,2], Yiyang Feng[1,2], Guilin Yang[1,2,*], Wenjun Shen[1,2], Silu Chen[1,2], Tianjiang Zheng[1], Junjie Li[1]

Abstract— **Industrial robots and collaborative robots are widely employed in industry and are progressively being utilized to assist individuals in their daily routines. To improve their absolute accuracy, self-calibration methods using portable local measurement devices are cost-effective solutions. However, compared with the conventional external calibration methods, self-calibration methods employing two configurations as a calibration sample introduce more non-kinematic errors to the robot. Therefore, noise reduction is significantly necessary in self-calibration. A novel Piecewise-weighted Random Sample Consensus (RANSAC) method is proposed in this paper. Instead of choosing an optimal model with all inliers, the proposed method employs a general weight considering both the sample and hypothesis model qualities to generate a new model with Weighted Least Square (WLS) method. Besides, the proposed method turns the target of finding an uncontaminated set of inliers into the training of the proper weight coefficient for WLS, which not only improves the accuracy but also greatly enhances the speed. The self-calibration experiment on a 6 degree-of-freedom(DOF) robot CR10 shows that the accuracy of the proposed Piecewise-weighted RANSAC method makes a 27.7% accuracy improvement from that employing Least Square method, a 20.0% accuracy improvement from that employing standard RANSAC method, and a 5.5% accuracy improvement from that employing LO-RANSAC method. Besides, the proposed method is also over 10.9 times faster than the standard RANSAC method and 18.6 times faster than the LO-RANSAC method.**

I. INTRODUCTION

Industrial robots and collaborative robots have found extensive applications in industrial scenarios and are increasingly being utilized to serve people in everyday life. However, despite their significant potential in the commercial market, their accuracy remains a limiting factor in variety of fields. Conventional robot calibration, which employs the information of dozens of robot end-effector poses with their corresponding robot joint angles, is a mature approach widely employed by most of the industrial robot companies . However, to obtain the information of robot end-effector poses, it often needs an expensive external device with wide precise pose measuring ability, usually costs hundreds of thousands of dollars, such as a laser tracker or a precise measuring arm [1]–[3]. Therefore, robot self-calibration method based on geometric constraints has been proposed as an alternative choice, which only needs a cheap device with local precise sensing ability, often costs thousands of dollars or even less, such as Linear Variable Displacement Transducers (LVDTs) or cameras [4], [5]. Expect for the difference of the measuring devices, the information robot self-calibration required is also different from the conventional robot calibration method. The robot self-calibration employs relative pose information with two robot configurations, while conventional robot calibration method employs absolute pose information with only one robot configuration. Since the actual robot model are not completely coincident with the robot kinematic model, additional one robot configuration will introduce extra non-kinematic errors to the samples, making the kinematic model it calibrates having lower absolute positioning accuracy than that calibrated by the conventional method. Therefore, noise reduction is significantly necessary in robot self-calibration.

The most common robot calibration noise reduction method is Least Square Method, which is proved to be the best model estimation method when the noise has a normal distribution. Besides, Maximization Likelihood Estimation method is employed when the noise has other distributions. However, ideal calibration results of most of these statistic methods rely on a proper estimation of the noise distribution, which is usually a difficult task, especially when the collected samples are contaminated by some unpredictable large noise. Therefore, a robust estimation method named Random Sample Consensus (RANSAC) is proposed to eliminated the influence of these samples with large noise, and it is widely employed in the field of computer vision. The standard RANSAC method [6] selects a minimum required number of samples to train a hypothesis model. The samples whose distance to the hypothesis model smaller than the preset threshold θ are chosen to be the inliers for the hypothesis model. If the number of inliers is the largest so far, a new model will be trained with all these inliers. These iterations continue until the confidence that selected samples are all inliers is higher than η (usually η equals to 0.99 or more).

Plenty of RANSAC variants have been proposed, most of which are in the field of computer vision [7]–[13]. The improvements mainly focus on two directions, i.e. accuracy improvement and speed enhancement. For accuracy improve-

*Corresponding author

[1]Jianhui He, Yiyang Feng, Guilin Yang, Wenjun Shen, Silu Chen, Tianjiang Zheng, and Junjie Li are with Zhejiang Key Laboratory of Robotics and Intelligent Manufacturing Equipment Technology, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo, Zhejiang, China. hejianhui@nimte.ac.cn, fengyiyang@nimte.ac.cn, glyang@nimte.ac.cn, chensilu@nimte.ac.cn, zhengtianjiang@nimte.ac.cn lijunjie@nimte.ac.cn

[2]Jianhui He, Yiyang Feng, Guilin Yang, Wenjun Shen, and Silu Chen are with University of Chinese Academy of Sciences, Beijing, China.

ment, a method named Maximum likelihood Estimation Sample Consensus (MLESAC) is proposed by Torr et al [7] where they choose the model that maximize the likelihood rather than just the number of inliers. Their idea has been adopted by many later researchers. Chum et al [8] point out that the standard RANSAC has an incorrect assumption that a model with parameters computed from an outlier-free sample is consistent with all inliers. Therefore, to fix it, they propose a method called Locally Optimized RANSAC(LO-RANSAC) in which a local optimization will be run to get a better hypothesis model once a new maximum number of inliers has occurred. Barath et al [9] propose a method called Graph-cut RANSAC, where they employ a local optimization method with a consideration of both the distance of the samples to the model and the spatial coherence of the inliers and the outliers. Another practical method proposed by Barath et al [10] called Marginalizing Sample Consensus (MAGSAC) avoids the introduction of the inlier-outlier threshold into the algorithm manually. An assumption of chi distribution for the distance of samples to the hypothesis model has been made in MAGSAC. The optimal Model is obtained by a Weighted Least Square (WLS) method, where the weight comes from the marginalization over the noise standard deviation σ of the point likelihoods of being inliers. As for speed enhancement, Matas et al [11] propose a simple but useful speed-up method named $T_{d,d}$ test, where a better model with all inliers will be trained unless all d randomly chosen samples are consistent with the hypothesis model. Further, Matas et al [12] propose another speed-up method called Sequential Probability Ratio Test (SPRT) based on Wald's theory, in which the likelihood ratio λ_j of a bad model over a good model is calculated by randomly select the test sample one by one. The test will not pass if λ_j is over a threshold A.

Though a great number of works has been done to improve the standard RANSAC method, we find most of them only consider the best hypothesis model while abandon all the other hypothesis models, which is a great waste of computation resources. In our opinion, this phenomenon is due to the characteristic of computer vision applications, where the samples are numerous with both high-quality and low-quality data. Therefore, the main concern in computer vision is to find the best samples. However, it is different in the field of robot calibration where the number of samples are limited and generally most of them are ordinary data (neither very good nor very bad). Hence, the main concern of noise reduction in robot calibration is to maximize the utilization of information from the limited number of samples available.

In this paper, a Piecewise-weighted RANSAC method has been proposed to fully utilize the sample and model information in each iteration of RANSAC. A general sample weight considering both the sample weight for each iteration and the hypothesis model weight for all iterations is employed for the WLS to generate a new model, which is thought to be better than a single optimal model in most cases. Besides, the proposed method changes the target from finding a model with all inliers to train a model with weights on given samples, making it possible to stop the iteration early. Therefore, the algorithm can be greatly accelerated. The simulations and experiments conducted on a 6-DOF robot CR10 validates the high-accuracy and high-speed of the proposed Piecewise-weighted RANSAC method over the standard RANSAC and LO-RANSAC.

The main contributions of this paper are as follows:

- a thought of utilizing all abandoned models is first introduced to the RANSAC method;
- a Piecewise-weighted RANSAC method considering both the hypothesis model weight and sample weight has been proposed;
- the algorithm is accelerated by changing the target from finding a model with all inliers to training the weights of giving samples;
- an introduction of RANSAC method to the field of robot self-calibration.

The following part of this paper is organized as follows. Section II introduces the proposed Piecewise-weighted RANSAC method in detail. Section III describes a two-step robot self-calibration method based on position and distance constraints, and the specific application details applying the proposed Piecewise-weighted RANSAC in the robot self-calibration method. Some simulations have been done in Section IV to compare the result of three different noise reduction methods. Section V demonstrates a self-calibration experiment on CR10. The conclusion is made in Section VI.

II. PIECEWISE-WEIGHTED RANSAC

The proposed Piecewise-weighted RANSAC method can be seen as a standard RANSAC method appended with an additional WLS method. In this section, the standard RANSAC is briefly reviewed, and the main considerations of weighting is introduced later, while the stopping criteria is demonstrated at last.

A. Standard RANSAC Method

The standard RANSAC method treats the samples as two types, i.e. inliers and outliers. To find the inliers out of given samples, a hypothesis model close to the actual model is needed for the distinction of the inliers and outliers. Therefore, a minimum required number set of samples is first randomly selected out of the given samples to generate the hypothesis model repeatedly, with a hope of finding a model which is trained by all inliers. Once the hypothesis model is determined, the distance of given samples to the actual model can be estimated. Then, the inliers and outliers are distinguished by a preset threshold θ. The hypothesis model is retrained by all the inliers if the number of inliers is most so far. The iteration will keeps until the confidence that selected samples are all inliers is higher than η. The number of iterations required is

$$N = \frac{log(1-\eta)}{log(1-\epsilon^m)}, \tag{1}$$

where m is the minimum number of samples required for the training of a hypothesis model, ϵ is the fraction of inliers in the given samples.

It is easy to see from (1), with the increment of m and the decrement of ϵ, N increases exponentially, making the computation time unbearable long. Fortunately, this stopping criteria is abandoned in the Piecewise-weighted RANSAC, replaced with a new stopping criteria which will be introduced later in this section.

B. Main Considerations of Weighting

Instead of treating the given samples as inliers and outliers like most researchers did in the field of computer vision, each sample in Piecewise-weighted RANSAC is given a weight according to its quality. The general weight of a sample consists of two parts, i.e. sample weight for an iteration and hypothesis model weight.

The sample weight for an iteration reflects how good (uncontaminated by the noise) a sample is among all the samples in a given hypothesis model trained in the iteration. It depends on the distance of the sample to the hypothesis model. Supposing the hypothesis model is not far from the actual model, the samples with smaller distance from the hypothesis model is considered less contaminated by the noise. Hence, the samples with small distance to the hypothesis model deserves large weight. The weight of the sample i gained in the iteration j is calculated by the following equation

$$w_{i,j}^{s} = \left(1 - \frac{d_{i,j}}{\max_i d_{i,j}}\right)^{p}, \tag{2}$$

where $d_{i,j}$ is the distance of the sample i to the hypothesis model in the iteration j, and p is a constant positive number (other decreasing functions also work). Since the hypothesis models are different to each other in different iterations, a normalization has been made to eliminate the influence of the factor of hypothesis model quality.

The hypothesis model weight reflects how good a hypothesis model is among all hypothesis models. Different from the sample weight for an iteration calculated in every iteration, the hypothesis model weight is calculated after all iterations have been done, which depends on the number of the inliers it relates to. The weight of the trained model m_j gained in the iteration j is calculated by

$$w_{j}^{m} = \left(\frac{\left|I_j^m\right|}{\max_j \left|I_j^m\right|}\right)^{q}, \tag{3}$$

where $\left|I_j^m\right|$ is the number of the inliers related to hypothesis model in the iteration j, and q is also a constant positive number (other increasing functions also work).

The general weight of a sample is a comprehensive assessment of the sample considering its goodness in all randomized independent iterations. From the perspective of the statistics, if a sample is an inlier, it has higher probability to be a selection for the model training in an iteration. Conversely, if a sample is selected for more times, it is more likely to be an inlier. Therefore, the general weight of a sample should be a sum of all the weights the sample gained in the iterations. While the weight a sample actually gained in an iteration depends on the goodness of the sample in the given hypothesis model and the quality of that hypothesis model, which are exactly the sample weight for an iteration and the hypothesis model weight mentioned above. Since these two factors are coupled, a multiplication of these two weights is chosen to assess the goodness of a sample in a single iteration. Therefore, the general weight of the sample can be represented by

$$w_i = \sum_{j=1}^{n} w_{i,j}^{s} w_j^{m} \tag{4}$$

where n is the number of iterations.

C. Stopping Criteria

The stopping criteria of standard RANSAC is to find a set of samples with all inliers, which is a very strict condition. With the number of model parameters increases and the fraction of inliers decreases, the required number of iterations increases exponentially. However, in the proposed Piecewise-weighted RANSAC, things are completely different. The Piecewise-weighted RANSAC method is not just a trivial weighted method, but changes the target of RANSAC from finding a set of samples with all inliers to training a model with weights on given samples. In this point, the iterations in standard RANSAC are no longer separated entities, but an organic whole which serves to train a weighted model. ”The given samples with weight in each iteration” itself serves as ”a sample” to train the weighted model. The more iterations it goes, the more ”samples” it has to train the required weighted model.

From this point of view, according to the Law of Large Number, the general weight of a sample in the proposed Piecewise-weighted RANSAC will approach to the actual weight it supposes as iteration goes on, like many machine learning methods do. Thus, the iteration can be stopped once the increment of the number of inliers goes flat. To this point, a period evaluation of the weighted model in the proposed Piecewise-weighted RANSAC is needed. After an iteration of t times, the general weight of each sample should be generated for a weighted model and the number of inliers need to be calculated. The stopping criteria of this method is

$$\left|I_k^M\right| \leq \left|I_{k-1}^M\right|, k = \frac{j}{t} \tag{5}$$

where k is the number of generations of the weighted model, $\left|I_k^M\right|$ is the number of inliers related to the k^{th} generated weighted model.

III. ROBOT SELF-CALIBRATION

This section first introduces a two-step robot self-calibration method based on position and distance constraints [4]. Then, the incorporation of the proposed Piecewise-weighted RANSAC method into robot self-calibration is demonstrated.

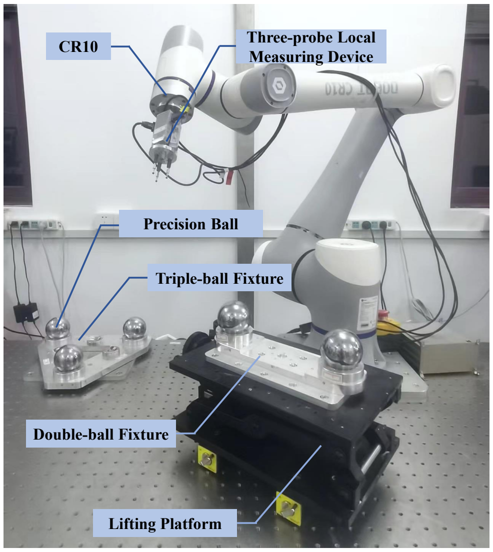

Fig. 1: A robot self-calibration device employing position and distance constraints

A. A Two-step Self-calibration Method based on Position and Distance Constraints

In our previous work [4], a novel robot self-calibration device has been developed which mainly includes five parts, i.e. a three-probe local measuring device, a double-ball fixture,a lifting platform, five precision balls, and a triple-ball fixture, as shown in Fig. 1.

The three-probe local measuring device is applied to calculate the position of the precision balls represented in the base frame. It contains three LVDTs, whose readings can be applied to calculate the position of a precision ball by touching the target precision ball with itself. This position of precision ball calculated by LVDTs is represented in the end-flange frame located on the three-probe local measuring device, which is coincide with the tool frame of the robot. Further, employing nominal robot kinematic model with readings of joint angles, the positions of the precision balls can be represented in the base frame.

The double-ball fixture with two precision balls is employed for realizing position and distance constraints. By touching the same ball on this fixture twice, because of the existing error between the nominal robot model and the actual robot model, two different sets of data usually generate two different positions. The error can be employed to identify the actual robot model, and forms a sample of position constraint. On the other hand, by touching different balls on this double fixture, the distance between the calculated positions by nominal robot model is also different from the actual distance of two precision balls. Thus, this distance error can also be employed to identify the actual robot parameters, and these two sets of data forms a sample of distance constraint.

The lifting platform is used to move the position of the double fixture in the robot workspace, which makes it have the ability to collect position and distance constraints information in a wide range space.

The triple ball fixture with three precision balls is employed for base calibration sample collection. Different from the double-ball fixture, the triple-ball fixture is fixed on the experimental platform. The positions of these three precision balls are combined to form a world frame (as a reference). By touching these three balls several times with three-probe measuring device, the position error between nominal positions and actual positions (base calibration sample) can be utilized to calibrate the base frame of the robot.

The self-calibration method consists of two steps, i.e. robot arm calibration step and base calibration step. The robot arm calibration step employs the samples with position and distance constraints to identify the errors of n-DOF robot joint twists (the twist errors)which are in the form of $SE(3)$. Since $SE(3)$ is a connected matrix Lie group, the twist errors $\delta\mathbf{T}_i \in SE(3)(i = 1, 2, ...n)$ can be represented in the form of ([14], Corollary 3.26)

$$\delta\mathbf{T}_i = e^{\delta\mathbf{t}_i^1} e^{\delta\mathbf{t}_i^2} ... e^{\delta\mathbf{t}_i^l}, \tag{6}$$

where $\delta\mathbf{t}_i^1$, $\delta\mathbf{t}_i^2$, ..., $\delta\mathbf{t}_i^l \in se(3)$ can be iteratively identified for l times by a modified Least Square method

$$\mathbf{X} = \left(\mathbf{A}_h^T\mathbf{A}_h + \alpha_1\mathbf{I}\right)^{-1} \mathbf{A}_h^T\mathbf{Y}_h, \tag{7}$$

where $\mathbf{X}$ is a matrix containing all errors of robot joint twists represented in a form of $se(3)$, $\mathbf{Y}_h$ is a matrix containing all sample errors of the position and distance constraints, and $\mathbf{A}_h$ is a Jacobian matrix between robot joint twist parameter errors and sample errors. Here, the additional term $\alpha_1\mathbf{I}$ is for singularity avoidance. α_1 is a tiny value which can be assigned as the smallest positive singular value of $\mathbf{A}^T\mathbf{A}$. This sigularity avoidance method is usually called Tikhonov regularization. [15]

The base calibration step is after the robot arm calibration step, where the error of the robot base frame relative to the frame formed by three precision balls is identified by the same method, i.e.

$$\mathbf{X}_b = \left(\mathbf{A}_b^T\mathbf{A}_b + \alpha_2\mathbf{I}\right)^{-1} \mathbf{A}_b^T\mathbf{Y}_b, \tag{8}$$

where $\mathbf{X}_b$ is a matrix containing the robot base frame error represented in a form of $se(3)$, $\mathbf{Y}_b$ is a matrix containing the all sample errors between the actual position of precision balls and the nominal position of that calculated by forward kinematics, and $\mathbf{A}_b$ is a Jacobian matrix between robot base frame errors and sample errors. Since $\mathbf{A}_b$ is column full rank, α_2 is set to 0 in this step.

B. Application of Piecewise-weighted RANSAC on Robot Self-calibration

Since the reference information for base calibration is the absolute position, which is actually similar to the conventional external robot calibration method with little noise, the base calibration still employs the iterative Least Square method mentioned above. The Piecewise-weighted RANSAC is only applied in the first step of self-calibration, i.e. robot

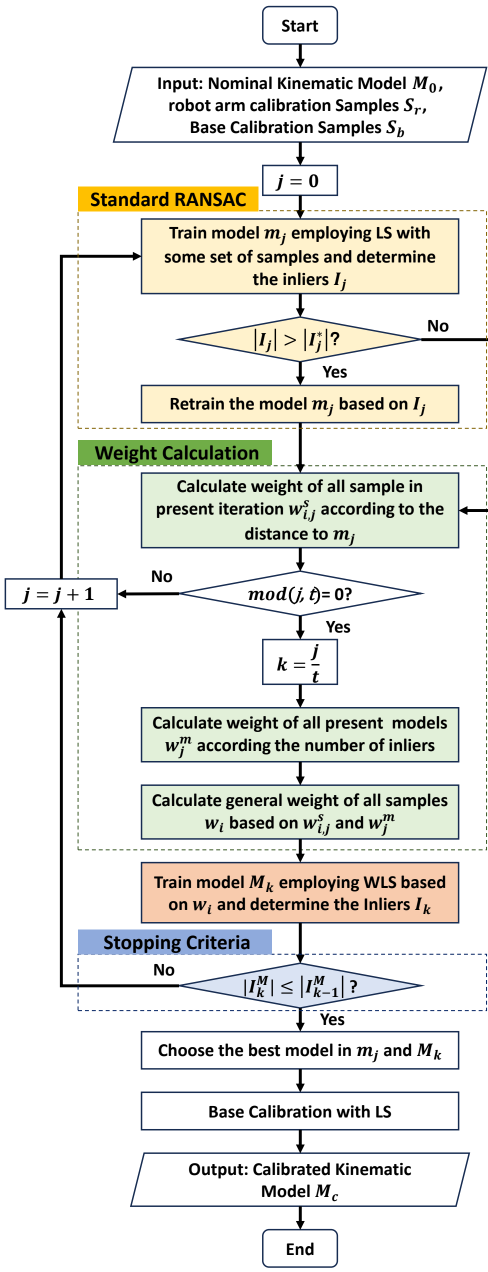

Fig. 2: Flowchart of the robot self-calibration employing Piecewise-weighted RANSAC

arm calibration step, whose calibrated accuracy accounts for a significant proportion in robot calibration.

The flowchart of the proposed Piecewise-weighted RANSAC method appplied on the robot self-calibration is shown in Fig. 2. First, as in standard RANSAC, several samples are needed to be randomly selected for a hypothesis model generation. However, different from the standard RANSAC, who requires the minimum number of samples to make the iterations as less as possible, this constraint is not mandatory in the Piecewise-weighted RANSAC. Actually, in practical robot calibration procedure, because the possibility of singularity, the data selected for calibration is usually a little larger than the required $4r + 2p + 6$ minimum number [2]. After the generation of a hypothesis model by these selected samples, the inliers and their sample weights for the iteration are determined. Once the inlier number related to the hypothesis model is the best so far, a new train of the hypothesis model is required, denoted as m_j Then, as these iterations repeats, when the number of iterations reaches a multiple of the preset number t, an assessment of all the hypothesis models is required, with the calculation of the hypothesis model weight. Meanwhile, the general weights of samples are calculated from the combination of the sample weight for each iteration and corresponding hypothesis model weight obtained before. Further, a weighted model is derived by Iterative WLS method, denoted as M_k. The iteration of first step calibration goes on until the number of inliers found related to the weighted model generated at k^{th} time M_k is no more than before. Note that $\left|I_k^M\right| < \left|I_{k-1}^M\right|$ may occur because of overfitting. Lastly, a base calibration is needed based on the best model in m_j and M_k to derive the final calibrated robot kinematic model M_c.

IV. SIMULATION

Computer simulations are conducted on a self-calibration case of a 6-DOF collaborate robot CR10 in this section to evaluates the performance of the proposed Piecewise-weighted RANSAC, with comparison to the Least Square Method, the standard RANSAC method, and the LO-RANSAC method.

A. Simulation Data Preparation

For simulation purpose, kinematic model parameters, samples for robot arm calibration step, samples for base calibration step and test samples are generated as follows.

The kinematic model provided by the Dobot company is applied for the nominal kinematic model as the initial guess for robot self-calibration, while the actual kinematic model for the generation of the samples for two calibration steps is obtained by adding small errors to the parameters of nominal kinematic model. The nominal and actual kinematic model are both based on the Local Product of Exponential (LPOE) formula [16].

The samples for robot arm calibration step consists of two parts, i.e. position constraint samples and distance constraint samples, with joint angle data $\mathbf{q}_r$ and corresponding precision ball position data represented in the robot tool frame $\mathbf{P}_r^{tool}$ (equivalent data as LVDT readings). The actual distance for distance constraint, i.e. the distance between two precision balls, is set as $200mm$. These two types of samples can be generated separately, but it will be more efficient to generate them at the same time. In fact, by touching each of two balls on the double-ball fixture 3 times, 6 position constraint sample and 9 distance constraint sample can be generated. We denote them as a group of samples. In our simulation, 25 groups of samples are randomly chosen in the workspace of the robot, where 12 groups of samples are given with noise of $N(0, 0.2mm)$ added on each vector component of $\mathbf{P}_r^{tool}$, while the other 13 groups of samples are given with noise of $N(0, 0.8mm)$.

The samples for base calibration step also contain joint angle data $\mathbf{q}_b$ and corresponding precision ball position data represented in the robot tool frame $\mathbf{P}_b^{tool}$. Totally 9 samples are generated for base calibration, 3 for each of the precision ball on triple-ball fixture. The specific coordinates of these three precision balls in world frame are $(200mm, 100mm, 0)$, $(250mm, -100mm, 0)$, and $(400mm, 0, 0)$, respectively. The noise of $N(0, 0.2mm)$ are added on each vector component $\mathbf{P}_b^{tool}$ the samples for base calibration step.

100 samples with randomly generated joint angle data and corresponding position of robot end-effector represented in the world frame are generated as test samples. Of course, the corresponding positions of robot end-effector are calculated by forward kinematics based on the actual kinematic model of the robot.

B. Comparison of Four Noise Reduction Methods

The same robot self-calibration simulation case with data generated above has been made by employing four noise reduction methods, i.e. Least Square method, standard RANSAC method, LO-RANSAC method and Piecewise-weighted RANSAC method. The average position error of the 100 test samples calculated by the calibrated model employing LS method is $0.91mm$, as shown in Table I.

Before the discussion of the results employing other methods, some preset conditions are necessary to be demonstrated.

Since there are random factors included in the latter three methods, the simulations which employ RANSAC method, LO-RANSAC method and Piecewise-weighted RANSAC method are all simulated for three times. Meanwhile, considering the fairness of the comparison between these two methods, the samples randomly chosen in the simulations are set as the same by employing the ”rng” function in MATLAB 2023a.

For the preset parameters of RANSAC, the threshold of inlier θ is set to $0.5mm$ which is a desired robot positioning accuracy for both position and distance constraint samples. The number of samples required for initial model generation is set as 18, 9 for position constraint with 3 parameters for each sample and 9 for distance constraint with 1 parameter for each sample, totally 36 parameters which satisfy the minimal number required for a 6-DOF robot. The confidence of the RANSAC method is set to 0.99. The average position error of the same 100 test samples calculated by the calibrated model employing RANSAC method is $0.70mm$, which is less than that employing LS method.

The overlapped preset parameters of LO-RANSAC method with the RANSAC method are the same as demonstrated above. The additional preset parameters are introduced as follows: the threshold multiplier K is set to $\sqrt{2}$, the size of inner sample for position constraint and distance constraint are both set to $\min(18, \frac{I_j^m}{2})$, the inner sampling repetitions N_{rep} is set to 10. The average position error of the same 100 test samples employing LO-RANSAC method is $0.62mm$, which is less than that employing LS method and standard RANSAC method.

As for the preset parameters of Piecewise-weighted RANSAC, considering the balance of the time of calculation and the accuracy of the calibrated result, a weighted model is generated for every 100 iterations, i.e. $t = 100$. Besides, the constant weight coefficients p and q which is a mean for weight-assign strategy adjustment, plays a significant role in Piecewise-weighted RANSAC, while may be different from case to case. Therefore, a simulation has been made to better understand how p and q influence the calibration result. The calibration result for Piecewise-weighted RANSAC is shown in Fig. 3. As the result shows, the average position error first tends smaller, while then goes larger as p and q increase. The smallest average position error employing Piecewise-weighted RANSAC in this simulation case is $0.419mm$ with p and q equals to 2 and 50, respectively, which is a much better result than that employing previous three methods. But it is still a little worse than the external calibration result by employing laser tracker with the method proposed in [16], which is $0.37mm$.

To further understanding the simulation result, a deeper analysis is required. Actually, LS and RANSAC method can be seen as special cases of Piecewise-weighted RANSAC method by employing different weighting functions. The LS method can be seen as a special case of Piecewise-weighted method with $p = 0$ and $q = 0$. The sample weight for an iteration employing RANSAC method can be seen as a step function where it is 1 when the normalized distance of samples to the hypothesis model is less than the given threshold $\bar{\theta}$ (which is also normalized). While the hypothesis model weight of RANSAC can be seen as 0 except for the best model equals to 1. This can be treated similar as a special example of Piecewise-weighted method with $q \to \infty$.

In practical applications, the users can employ the best weight coefficients p and q by simulating with several different parameters and choosing the case with most inliers. This is relatively easy since the choice for a not-bad calibration result has a relative wide range choice for both p and q, as shown in Figure 3.

V. Experiment

In this section, experimental studies are conducted on CR10 to investigate the performance of the proposed Piecewise-weighted RANSAC in practical robot self-calibration applications.

Prior to the experiment, the three-probe local measuring device is calibrated first as stated in [4]. Besides, the positions of the five precision balls are measured by GOM ATOS III Triple Scan (GOM). The distance between two precision balls on the double-ball fixture is $200.19mm$, and the coordinates of three precision balls on the triple-ball fixture are $(0, 0, 0)$, $(200.04mm, -0.01mm, 0.10mm)$, and $(100.02mm, 173.21mm, 0.08mm)$ respectively.

During the experiment, 35 groups of measurement data are collected for the robot arm calibration step with 4 distance

TABLE I: The performance of different noise reduction methods employed in the self-calibration simulation of CR10 compared with laser tracker based calibration method

Method	Least Square	Standard RANSAC (3000 iterations)	LO-RANSAC (3000 iterations)	Piecewise-weighted RANSAC	Laser Tracker-based (external calibration)
Runtime (unit: s)	0.15	218.87	342.36	**17.32**	0.16
Average Position Error of Test Samples (unit: mm)	0.91	0.70	0.62	**0.42**	0.37

TABLE II: The performance of different noise reduction methods employed in the self-calibration experiment of CR10 compared with laser tracker-based calibration method

Method	Least Square	Standard RANSAC (3000 iterations)	LO-RANSAC (3000 iterations)	Piecewise-weighted RANSAC	Laser Tracker-based (external calibration)
Runtime (unit: s)	0.12	170.55	290.27	**15.64**	0.12
Average Position Error of Test Samples (unit: mm)	0.72	0.65	0.55	**0.52**	0.31

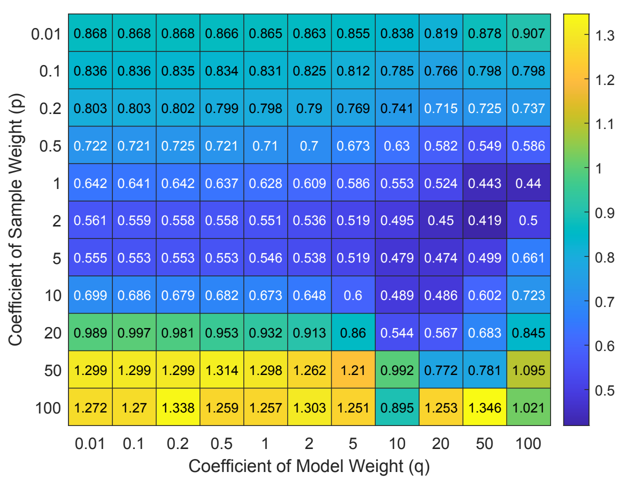

Fig. 3: Average position error for a 6-DOF robot CR10 self-calibration employing Piecewise-weighted RANSAC method with different coefficients of sample weight and model weight

samples and 2 position samples in a group, and 6 samples have been collected for the base calibration step. Besides, another 20 samples are collected for test. The test sample is collected by employing the three-probe local measuring device connected to the robot end to touch some precision ball whose actual position is measured by GOM and laser tracker. 10 position randomly chosen in the workspace of CR10 are employed for test, each position with 2 samples.

The position errors of the test samples between the actual position of the precision ball, and the position calculated by the calibrated model employing Least Square method, standard RANSAC method, LO-RANSAC, and Piecewise-weighted RANSAC method are shown in Fig. 4. Meanwhile, the position errors with the model calibrated by conventional external robot calibration method with laser tracker is also employed as a reference. The average position errors for 20 test points calculated by the self-calibration mentioned above are $0.72mm$, $0.65mm$, $0.55mm$ and $0.52mm$, respectively. Though the self-calibration accuracy employing Piecewise-weighted RANSAC method is still far from the conventional external calibration accuracy employing laser tracker, which is $0.31mm$ in our test, the accuracy makes a 27.7% improvement from that employs Least Square method, 20.0% improvement from that employs standard RANSAC method, and 5.5% improvement from that employs LO-RANSAC method.

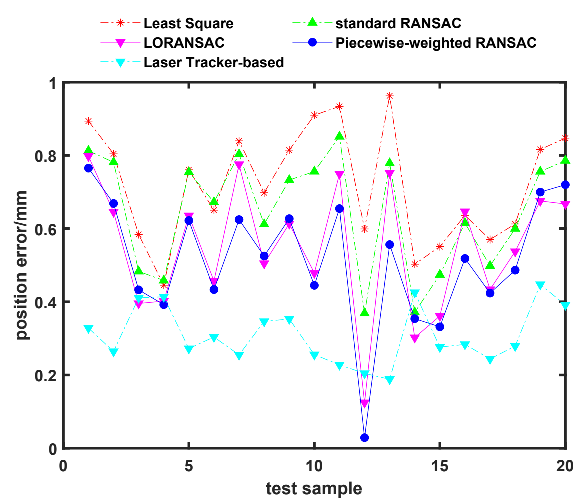

Fig. 4: The position error of 20 test samples based on the calibrated model obtained by different noise reduction methods

The runtime of the programs by different noise reduction methods is also considered. All programs are run on Matlab 2023a, with an Intel Core i5-13600KF CPU. As shown in Table II, the method employs Least Square runs for only $0.12s$, which is the fastest among three methods. Meanwhile, since the method employing the proposed Piecewise-weighted RANSAC method with new stopping criteria stops early, it runs for $15.64s$. Comparing the runtime of the method employing standard RANSAC which runs for $170.55s$ and LO-RANSAC method which runs for $290.27s$, the proposed method is 10.9 times faster than that employing the standard RANSAC method and 18.6 times faster than that employing LO-RANSAC method.

VI. CONCLUSIONS

In this paper, a novel modified RANSAC method named Piecewise-weighted RANSAC has been proposed for a new application of robot self-calibration. Different from the applications in computer vision, the samples obtained in robot self-calibration is few but with relatively good qualities. Therefore, the focus of the noise reduction in robot self-calibration is to utilize the limited samples efficiently rather than just abort the outliers, which will probably decrease the general accuracy of the calibrated robot in a wide workspace. To achieve this point, the proposed Piecewise-weighted RANSAC method weights not only all the samples for every iteration according to the distance to the hypothesis model, but also weights every hypothesis model according to the number of inliers, deriving a final general sample weight considers both of these two weights to generate a new model with the method of WLS. Besides, due to the consideration of employing all the hypothesis models with different weights, the target of finding an uncontaminated set of inliers in RANSAC, changes to the training of the proper weight coefficient for WLS. This makes the proposed Piecewise-weighted RANSAC method can be stopped immediately once the increment of the number of inliers goes flat, which saves lots of time. A self-calibration experiment result on a 6-DOF robot CR10 shows that the proposed Piecewise-weighted RANSAC method makes a 27.7% accuracy improvement from the Least Square method, a 20.0% accuracy improvement from the standard RANSAC method and a 5.5% accuracy improvement from the LO-RANSAC method. Besides, the proposed method is also over 10.9 times faster than the standard RANSAC method and 18.6 times faster than the LO-RANSAC method.

Though the proposed Piecewise-weighted RANSAC method performs well in the application of robot self-calibration, there still exists some aspects that can be improved. For example, the choice of the hyperparameters p and q are chosen by a trial method, which is not efficient. Besides, the core idea of the Piecewise-weighted RANSAC method which utilized the abandoned hypothesis model information efficiently can be employed to the other RANSAC method, which might have a better performance with a proper treatment.

ACKNOWLEDGMENT

This research was funded by the National Key Research and Development Program of China (Grant Number: 2022YFB4702502), the National Natural Science Foundation of China (Grant Number: U22A20177, 92048201, U20A20282), and Ningbo Science & Technology Bureau (Grant Number: 2021Z128, 2023Z135).

REFERENCES

[1] A. Nubiola and I. A. Bonev, "Absolute calibration of an abb irb 1600 robot using a laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 236–245, 2013.

[2] T. Sun, B. Lian, S. Yang, and Y. Song, "Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 816–834, 2020.

[3] X. Yang, L. Wu, J. Li, and K. Chen, "A minimal kinematic model for serial robot calibration using poe formula," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 326–334, 2014.

[4] J. He, L. Gu, G. Yang, Y. Feng, S. Chen, and Z. Fang, "A local poe-based self-calibration method using position and distance constraints for collaborative robots," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102685, 2024.

[5] R. Wang, A. Wu, X. Chen, and J. Wang, "A point and distance constraint based 6r robot calibration method through machine vision," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101959, 2020.

[6] K. G. Derpanis, "Overview of the ransac algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[7] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[8] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*. Springer, 2003, pp. 236–243.

[9] D. Barath and J. Matas, "Graph-cut ransac," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6733–6741.

[10] D. Barath, J. Matas, and J. Noskova, "Magsac: marginalizing sample consensus," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 197–10 205.

[11] J. Matas and O. Chum, "Randomized ransac with td, d test," *Image and vision computing*, vol. 22, no. 10, pp. 837–842, 2004.

[12] ——, "Randomized ransac with sequential probability ratio test," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1727–1732.

[13] E. Brachmann and C. Rother, "Neural-guided ransac: Learning where to sample model hypotheses," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4322–4331.

[14] B. C. Hall, "An elementary introduction to groups and representations," *arXiv preprint math-ph/0005032*, 2000.

[15] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method." *Sov Dok*, vol. 4, pp. 1035–1038, 1963.

[16] I.-M. Chen, G. Yang, C. T. Tan, and S. H. Yeo, "Local poe model for robot kinematic calibration," *Mechanism and Machine Theory*, vol. 36, no. 11-12, pp. 1215–1239, 2001.