Available online at www.sciencedirect.com
SCIENCE DIRECT®
Mechanism and Machine Theory 41 (2006) 929–944
Mechanism and Machine Theory
www.elsevier.com/locate/mechmt

Interval method for calibration of parallel robots: Vision-based experiments

David Daney [a,*], Nicolas Andreff [b], Gilles Chabert [a], Yves Papegay [a]

[a] COPRIN Research Group, INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
[b] LaMI/LASMEA – IFMA/UBP/CNRS, Institut Français de Mécanique Avancée, 63175 Aubière, France

Abstract

This paper is a theoretical and experimental study of how interval arithmetic and analysis methods can be used to achieve (1) numerical certification of the kinematic calibration of parallel robots, and (2) a possible validation of the kinematic model used in calibration. First, a detailed description is given of our experimental device and vision-based measurement method. The usual calibration methods are then reviewed and applied to our experimental data set, yielding a motivation for numerical certification of the results. Next, interval calibration methods (which have already been described in a previous work) are also reviewed and applied to the data. Finally, the experimental results are discussed and interpreted.

Keywords: Parallel robots; Calibration; Certification; Vision-based metrology; Interval methods

1. Introduction

Parallel manipulators are an appealing solution to many applications, thanks to their highly accurate positioning and orientation. Such accuracy relies, however, on a robust and accurate calibration of the physical robot configuration. This is a difficult task, both theoretically and practically, but not a critical one as calibration can be performed off-line. A robot configuration (composed of its end-effector pose and joint values) is linked to its kinematic parameters by the equations of the kinematic model. Calibration is achieved by measuring several robot configurations, each of them providing a set of equations involving the kinematic parameters (the calibration equations), and then solving the entire system. This implies that the number of equations given by the measurements has to be at least as large as the number of unknown parameters. Since the measurement data are usually given by a sensor, however, it is also necessary to take into account the noise associated with this device. To this purpose, the sensitivity of the calibration to data uncertainties is reduced by measuring additional configurations so that the number of equations in the system is larger than the

* Corresponding author.
E-mail addresses: David.Daney@sophia.inria.fr (D. Daney), Nicolas.Andreff@ifma.fr (N. Andreff), Gilles.Chabert@sophia.inria.fr (G. Chabert), Yves.Papegay@sophia.inria.fr (Y. Papegay).

0094-114X/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechmachtheory.2006.03.014

number of unknown parameters. Hence, the resulting system of equations is non-linear and *over-determined*. The classical approach to solving such a problem is through a non-linear least squares optimization method. A least squares method can only guarantee that the sum of the residuals is minimized, however, not that the calibration equations will be satisfied even within the margin provided by uncertainties. Neither can it guarantee that the accuracy of the robot will be improved over the whole workspace after calibration. Post-processing is therefore necessary in practice to validate the calibration results. Unfortunately, this step can be very costly as is the case for Gough platforms [1].

Some improvements to the least squares method have been proposed, such as providing a quality index for each solution in situations where the data uncertainties can be modeled [2]. This may be done if the distribution of the measurement error is known (e.g. Gaussian noise). This noise model may be difficult to obtain for some measurement devices, however, as well as in the presence of mechanical constraints on the calibration. Furthermore, such an approach allows one to obtain at best a probabilistic result.

We have proposed in [3] another approach, whose objectives are quite different from the least squares method. The main idea is that the result of a calibration method should give kinematic parameter values that satisfy the calibration equations. As uncertainties affect these equations, however, it is not possible to obtain unique values for each parameter. Our approach, based on an interval analysis version of the so-called implicit calibration method [4–6], allows one to obtain ranges for the kinematic parameters that are guaranteed to include all values that satisfy the calibration equations whatever the real values of the uncertain measurement data may be. The principle objectives of the two approaches may be summarized as follows:

- *least squares*: to obtain one solution that is the best fit to the whole system of calibration equations, while accepting that some calibration equations may not be satisfied for all configurations.
- *our approach*: to obtain a range of solutions that is guaranteed to include the parameter values that will satisfy the calibration equations.

We present in this paper an original experimental platform, a complete set of experiments testing this new method, and an extensive comparison with the results of classical calibration methods.

One of the key issues in effective calibration is finding an appropriate sensor to measure the robot pose. Ideally, the sensor should be easy to set up and have low cost. This device should also be able to sense the whole workspace, since it has been reported several times [7,8] that calibration over larger workspaces leads to greater accuracy. These arguments point to vision-based sensors as a perfect solution.

Vision has already been proposed for the kinematic calibration of serial mechanisms [9,10]. It enables one to achieve sufficiently precise measurements of the robot pose over a large area, using a low-cost sensor. Vision has therefore been considered as a potential tool for the kinematic calibration of parallel mechanisms [11]. An implementation has even been proposed in [12] for the Delta-like H4 robot, where vision-based measurements achieved a repeatability on the order of 10 μm.

Since this paper is mainly concerned with experiments and the study of experimental results, we deliberately open with a section devoted to the Experimental Calibration Problem (Section 2). In this section, basic modeling principles are recalled and the experimental set-up is described. Then focus is put on the processes of visual data collection and the statistical characterization of our experimental data. Section 3 is devoted to the application of classical calibration methods to the large data set. Although one obtains different numerical solutions depending on the method, it will be shown that these solutions have equivalent accuracy and physical relevance. This proves the need for a certified interval including all feasible solutions, rather than a single solution. Section 4 introduces a novel method for obtaining such a certified solution set, and Section 5 validates the method experimentally using the vision-based data. Conclusions and perspective are given in Section 6.

2. Experimental calibration problem

This section describes the experimental issues involved in calibrating a Gough platform. The first subsection presents the theoretical aspects of the calibrating a Gough platform. In the second subsection our experimental robot device is introduced, and the final subsection presents the experimental measurement device in detail.

2.1. Gough platform calibration problem

A Gough platform [13,14], as depicted in Fig. 1, is used as a generic robot in this paper. This manipulator consists of two rigid bodies, known as the *base* and *mobile platforms*, connected by six prismatic actuators known as the *legs*. The legs are connected to the base and mobile platforms by spherical or universal joints.

The robot pose is given by a position vector P and a rotation matrix R; these parameters define the relationship between the frame $\Omega_c = (C, x_r, y_r, z_r)$ associated with the mobile platform and the fixed reference frame $\Omega_o = (O, x, y, z)$ associated with the base. The robot configuration ($\mathscr{X} = [P, R]$) depends on the length of each leg, as measured by a proprioceptive sensor. Each length is decomposed into the sum $l_i + L_i$, where l_i is the offset length (its initial length at configuration) and L_i the length variation measured by the sensor.

It can be shown that 23 parameters are required to fully model each leg [15]. To obtain a kinematic model of the robot (which is necessary for the calibration problem), it is mandatory to obtain values for all (23×6) parameters. This full model includes the kinematic parameters of the passive universal and spherical joints, but the calculation of so many parameters is highly time consuming. Hence, most authors assume that the universal and spherical joints are perfect and perfectly assembled in order to simplify the model. It may be thought that these are strong assumptions, but as shown in [16], the main source of error in positioning comes from limited knowledge of the joint centers and length offsets.

We will also make use of this simpler model, involving only the points of attachment A_i between the legs and base (with individual coordinates a_i expressed in the frame Ω_o), the points of attachment B_i between the legs and the mobile platform (with coordinates b_i expressed in the mobile frame Ω_c), and the offset lengths l_i for each leg i. This model requires only 42 parameters in all, seven for each leg. The corresponding *inverse kinematic* model expresses the length of the ith leg as follows:

$$\|P + Rb_i - a_i\|^2 = (L_i + l_i)^2. \tag{1}$$

In the case of the Gough platform the exact *forward kinematic* model is much harder to compute, but it can be shown that it is useless for calibration [17].

We will assume that for each of N selected calibration configurations, a measurement device (such as coordinate measurement machinery, theodolites, or in our case a vision device) delivers the position P_k and the orientation R_k. Additionally, internal sensors give the length variation of the legs $L_{i,k}$ for each configuration. As the legs are independent from the calibration point of view (i.e. the calibration equations obtained for one leg do not involve the kinematic parameters of the other legs), we separate the latter into six subproblems. We therefore simplify notation in the following discussion by omitting the leg index i.

For each subproblem, we thus have a vector of parameters $x = (a, b, l)$ and a list of measurements $M_k \in (M_1, \ldots, M_N)$, with $M_k = (P_k, R_k, L_k)$. We consider the function f defined by

$$f(x, M_k) = \|P_k + R_k b - a\|^2 - (L_k + l)^2. \tag{2}$$

The calibration problem can thus be expressed as a system of N inverse kinematic equations with the seven kinematic parameters a, b, l as unknowns:

$$f(x, M_k) = 0, \quad \text{for } k = 1, \ldots, N. \tag{3}$$

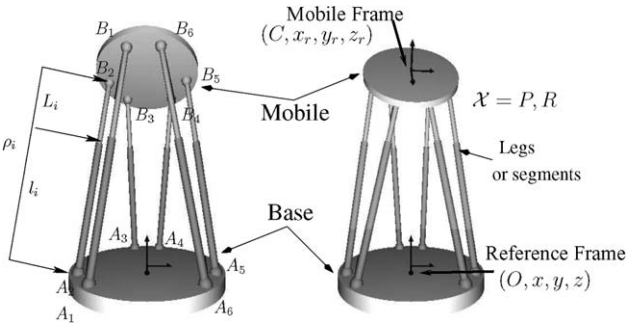

Fig. 1. Gough platform.

Table 1
Theoretical values of the kinematic parameters of the *Table of Stewart* in mm

Leg	a_x	a_y	a_z	b_x	b_y	b_z	l
1	−223.1792	−151.9573	0	−19.9938	−193.9723	0	345
2	223.1792	−151.9573	0	19.9938	−193.9723	0	345
3	243.1885	−117.3002	0	177.9818	79.6710	0	345
4	20.0093	269.2575	0	157.9880	114.3013	0	345
5	−20.0093	269.2575	0	−157.9880	114.3013	0	345
6	−243.1885	−117.3002	0	−177.9818	79.6710	0	345

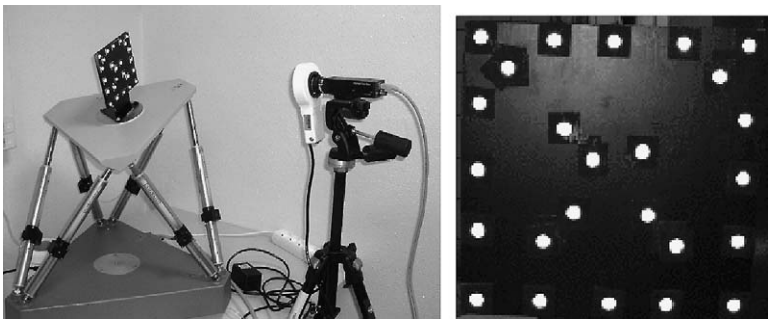

Fig. 2. Experimental set-up and visual target.

Such a system is well-determined if and only if $N = 7$, under-determined if N is smaller, and over-determined otherwise. Due to the noise in the measurements associated with the sensors, however, the solution computed for the well-determined problem may be experimentally meaningless and produce very bad results for different configurations. To reduce the effect of measurement noise, it is preferable to work with more equations than is minimally required.

Classical methods for solving over-determined systems are typically optimization-based, such as the one based on the analytic Jacobian given in [5]. Linearization [4] methods that produce an unique, optimized, least squares solution are also common. As we shall see later on, the techniques of interval analysis and constraint programming offer another type of solution as useful as these classical methods.

2.2. Experimental robot

The experimental platform used for testing and validating our calibration approach is built around a commercial DeltaLab *Table of Stewart* robot. This robot is a Gough platform as described above,[1] with a base of radius 270 mm, a mobile platform of radius 195 mm, and six legs whose length can vary between 345 mm and 485 mm.

The DeltaLab *Table of Stewart* was designed for academic and teaching purposes, so does not offer the precision and accuracy of an industrial robot. This makes the calibration process even more sensitive and necessary. Initial estimates (blueprint values) of the robot's 42 kinematic parameters $x = [a, b, l]$ are displayed in Table 1. Fig. 2 shows our experimental system: a 1024×768 CCD camera with a 4.2 mm objective, looking at a visual target attached to the mobile platform of the robot. This target is made of reflective white dots (to improve the accuracy) arranged in a special pattern. The placement of the dots is asymmetric, with diagonal points arranged to ease image analysis (see Section 2.3.1).

[1] Unlike the above description, all of the passive joints connecting the robot's legs to the base and mobile platforms are spherical joints. This change does not affect the simplified model.

It will be shown later how this vision device can be used to automatically collect data and associate an error distribution with the measurement. This information will be used to estimate an appropriate variation interval for the measurements.

2.3. Measurement device and data acquisition

A computer vision system was chosen to measure the mobile frame's position and orientation for the following reasons:

- This type of device is cheap and has already been proven sufficiently accurate for the calibration of parallel robots [12], with post-calibration positioning errors less than a millimeter. It is therefore well suited to the calibration of the Deltalab robot.
- Validating the interval approach for calibration requires an estimate of the measurement error distribution. One good experimental approach is to perform multiple measurements, which is only practical when the execution and computation times for the procedure are small and the procedure can be automated. The vision device meets both of these criteria.

2.3.1. Image detection

To obtain a statistical model of vision-based measurements requires a large number of images, many more than the usual number encountered in computer vision applications. It is therefore necessary to develop an efficient procedure for extracting measurements of the position and orientation from each image.

A method for accurate estimation of the position and orientation of a set of points with respect to the camera can be found in [18]. It was also proven in [19] that subpixel accuracy on the measurements (2/100 pixel) can be obtained using special patterns such as crosses and dots. A visual target was therefore made from such patterns and placed on the mobile platform.

It is intuitive to build a visual target by placing the patterns on a regular grid. Unfortunately, this results in a symmetric target that requires human interpretation (i.e. a human action such as a mouse click) to determine the starting point of the image detection. To help the human interpret the data an additional visual marker, not usually used for image detection, is often placed on the target to provide a unique interpretation of the regular grid. This method is useless in our case, since the large number of images does not allow for any human intervention.

Rather, an asymmetrical visual target is used that does not require any human interpretation. Using only image information, one should be able to place a reference frame on such a visual target. Moreover, this must be done relatively quickly so that a large number of images can be analyzed in a reasonable time.

One needs to build a target whose appearance is unique, no matter what position and orientation it is seen from. This brings us to the concept of image invariants, which are well known in the field of computer vision. The simplest image invariant is the *double ratio* (or cross-ratio) of 4 aligned points [20]. The double ratio signature is unique to a group of 4 points, and invariant under the rigid motion of these points in front of the camera. Using this signature it is therefore easy to compare two sets of 4 aligned points, and determine whether or not they are the same.

To find correspondences between a planar visual target and its image, it is necessary to define two axes on it. One may think at first that it is enough to have two alignments of 4 points with different double ratios, but this is not the case. Indeed, the double ratio is unsigned and thus does not discriminate between the two possible orientations of a 4-point alignment. Moreover, there may be a large number of 4-point alignments in the image, resulting in many possible combinations.

Consequently, we use (see Fig. 2) the two diagonal alignments of 7 and 6 points in order to cope with the physical dimensions of the patterns and target. These alignments should be chosen such that their first 4 points and last 4 points have different double ratios, which is sufficient to uniquely define the two axes. Once the axes are defined, the CAD model of the target can easily be used to extrapolate additional points that will improve the accuracy of the measurements.

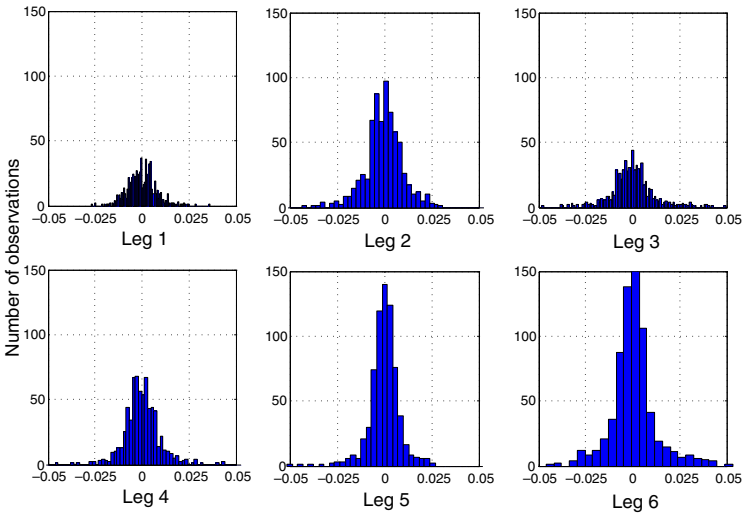

Fig. 3. Distribution of the errors on leg length (in mm).

Table 2
Standard deviation of the errors on the six leg length measurements

Leg number	1	2	3	4	5	6
Std (mm)	0.0077	0.0127	0.0117	0.0102	0.0118	0.0190

2.3.2. Data acquisition

Before starting the experiment the robot was measured in each of its $2^6 = 64$ extremal configurations, in which each leg was at either maximum (485 mm) or minimum (345 mm) extension. This set of configurations is named $\mathscr{X}_c$. These measurement configurations were chosen because the observability index[2] associated with an extremal pose is good for all Gough platforms [7]. In each configuration, 10 images were taken in order to quantify the repeatability of the 3D measurements. The average of these 10 images was also computed, yielding a total of 11 images for each configuration. This procedure was then repeated 11 times for each configuration, to quantify the repeatability of the robot positioning. This calibration set thus gathers a total of 11 (runs) $\times$ 64 (configurations) $\times$ 11 (images) = 7744 images.

An additional 64 configurations (named $\mathscr{X}_v$) were then chosen randomly inside the robot workspace, which will be used only to validate the calibration procedure. Each of these calibration poses also provided 10 data sets of 10 images, for a total of $10 \times 64 \times 11$ additional images.

Overall, a total of $(11 + 10) \times 64 \times 11 = 14{,}784$ images were taken. Approximately one day was required for the data collection, and it took 3 h to automatically analyze them.

In addition to the robot pose measurements, internal sensors recorded the six leg length measurements associated with each of the 11×64 configurations $\mathscr{X}_c$ and 10×64 configurations $\mathscr{X}_v$.

2.3.3. Evaluation of the measurement error distribution

2.3.3.1. Leg length measurement distribution. Fig. 3 shows the centered distribution of the six leg length measurements for all 11×64 configurations $\mathscr{X}_c$. Table 2 shows the standard deviation of the centered distribution.

2.3.3.2. Pose measurement distribution. Fig. 4 shows the centered distributions of the six independent components[3] of each of the $11 \times 64 \times 11$ images associated with the configuration $\mathscr{X}_c$, as computed by vision. Table 3

[2] The product of singular values, the condition number, the smallest singular value, or a linear combination of singular values of the identification Jacobian $\frac{\partial f}{\partial \kappa}$—see [8]—may be used as observability index.

[3] The orientation is decomposed here into three Euler angles only for illustration purposes. For the numerical computation, quaternions are preferred.

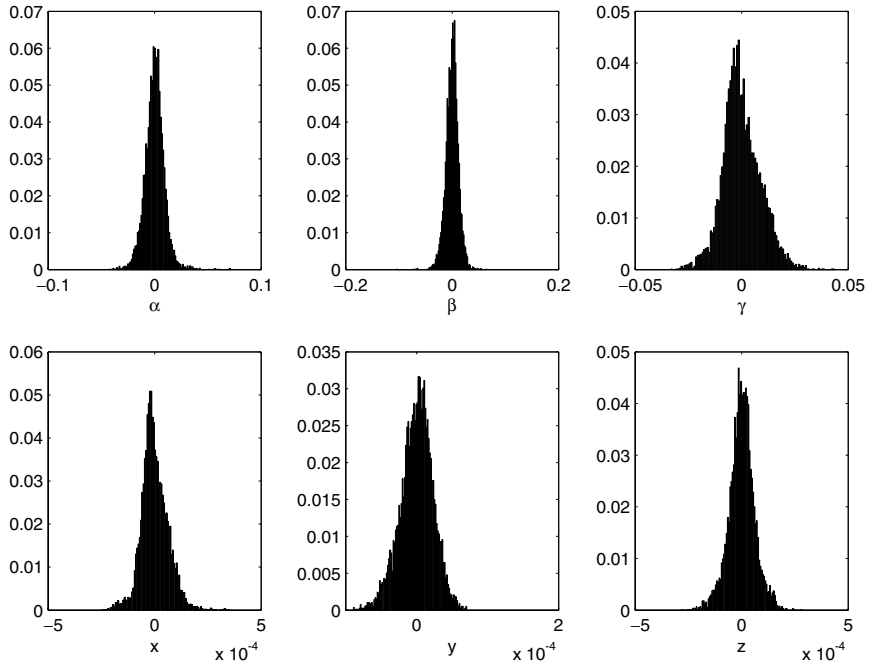

Fig. 4. Centered distributions of the 6 degrees of freedom of the $\mathscr{X}_c$ configurations, α, β, γ, Euler's angles (in degrees), x, y, z position (in m).

Table 3
Standard deviation of the centered distributions for the 6 degrees of freedom in the $\mathscr{X}_c$ configurations

Poses	α	β	γ	x	y	z
Std (degree, m)	9.23e−03	1.19e−02	8.94e−03	6.33e−05	2.30e−05	6.08e−05

shows the associated standard deviations. Note that since 11 images where taken for each configuration, the distributions take into account both measurement repeatability and robot repeatability.

3. Classical calibration

Experimental data is first fed into a least squares algorithm, providing us with a reference solution to the system of calibration equations. An accurate initial guess of the robot kinematic parameter values is needed. The result is described in the next section.

3.1. Initial estimation of kinematic parameters

The robot blueprint provides us with a good estimate of the leg length offsets l_i and the relative positions of the attachment points. The base and mobile frame positions are determined by fixing the $6 + 6$ values of the attachment points a_i, b_i. Typically we define the base frame Ω_a by fixing $a_{1x} = a_{1y} = a_{1z} = a_{2y} = a_{2z} = a_{3z} = 0$, and Ω_b by fixing $b_{1x} = b_{1y} = b_{1z} = b_{2y} = b_{2z} = b_{3z} = 0$. The other attachment point parameters are defined relative to Ω_a and Ω_b using the blueprint information.

The method proposed here relies on measurement of the robot pose $[P, R]$. This implies the use of an exteroceptive measuring device. However, any such device (e.g. laser tracking systems, theodolites, mechanical measuring devices, and cameras) delivers a target measurement (respectively of a reflective cube, reflective marks, physical interface, and visual target) with respect to its own reference frame. In order to obtain the configuration of the mobile frame with respect to the base frame from a exteroceptive sensor, one therefore needs to

know the rigid transformations between all three frames: the measuring reference frame, the robot base frame, and the robot mobile frame.

In this case study, the end-effector frame Ω_c is associated with the vision target and the reference frame Ω_o is associated with the camera frame.

The transformations between Ω_a and Ω_o and between Ω_b and Ω_c can be estimated using the so-called hand-eye calibration techniques [21–23]. In the present work the linear variant proposed in [24] is used. In this, the blueprint estimations of the attachment points are considered to be the same as their actual positions.

At the end of this process, the attachment points a_i and b_i are redefined in terms of Ω_o and Ω_c. Their values are used as an initial estimate for the non-linear least squares procedure.

3.2. Least squares method

There are two non-linear least squares optimization functions that may be associated with the calibration problem:

- $x_c = \min_x F^t F$ where $F = [f(x, M_1), \ldots, f(x, M_N)]$.
- $x_{c_\sigma} = \min_x F^t \Sigma_F^{-t} \Sigma_F^{-1} F$ where $F = [f(x, M_1), \ldots, f(x, M_N)]$, where $\Sigma_F^{-1} = \frac{\partial F}{\partial M} \Sigma_M^{-1}$ and $(\Sigma_M)_{i,i} = \sigma_M$ (see Section 2.3.3 for the determination of σ_M).

Both can be solved using the Levenberg–Marquardt method (as in the Matlab function *lsqnonlin*).

The initial kinematic parameter estimations are given in Table 1, redefined in the camera and target frame according to Section 3.1. The constraint equations $f(x, M)$ are those given in Section 2.1. The first solution x_c is a classical solution of the *inverse calibration method* for the Gough platform [5], while the second solution x_{c_σ} also takes into account measurement errors under the assumption of a Gaussian distribution. Typically, the sum of the square of the equation evaluation is minimized; in the solution x_{c_σ}, however, each equation is weighted by the covariance matrix Σ_F. This matrix models the influence of the measurement error distributions on the constraint equation. We provide an indirect, first-order approximation to Σ_F as a function of the covariance matrix associated with measurement Σ_M (Section 2.3.3) and the Jacobian matrix $\frac{\partial F}{\partial M}$ constructed from the derivatives of the constraint equations $f(x, M)$ with respect to the measurement vector M [2].

3.3. Results and validation

Both classical algorithms converge in a few iterations, and correctly minimize the associated criteria. The solutions x_c and x_{c_σ} are presented in Tables 4 and 5, respectively. We should note that the results obtained are quite different in some cases.

To test the classical calibrations, we use x_c and x_{c_σ} to compute the positioning errors of the *validation configurations* $\mathscr{X}_v$ (Section 2.3.2). Table 6 presents the mean and the standard deviation of errors in the position and orientation, taken between the average measured position of each pose in $\mathscr{X}_v$ and the corresponding positions computed using the forward kinematic model and the parameters x_c or x_{c_σ}.

Fig. 5 presents results for the 64 validation configurations $\mathscr{X}_v$ in term of position/orientation improvement [100 × (error before calibration − error after calibration)/error before calibration]. For the sake of clarity, the

Table 4
Values of x_c (in mm)

Leg	a_x	a_y	a_z	b_x	b_y	b_z	l
1	−220.7358	−148.9040	−1.3908	−20.5951	−191.2210	0.2772	334.5216
2	220.1908	−148.9040	−1.3908	18.2577	−191.2210	0.2772	334.0422
3	240.5155	−116.3502	−1.3908	176.4621	78.6322	0.2772	337.5156
4	20.5843	264.2345	3.1014	156.3954	110.6810	−0.7735	333.2639
5	−18.3513	265.3592	3.9762	−155.1999	112.7371	−0.7923	334.7650
6	−242.2035	−115.4354	−2.9053	−175.3203	80.3917	0.7341	337.3083

Table 5
Values of x_{c_σ} (in mm)

Leg	a_x	a_y	a_z	b_x	b_y	b_z	l
1	−226.4666	−148.8772	7.1414	−24.6547	−191.3399	−5.5916	345.6908
2	226.7834	−148.8772	7.1414	15.9240	−191.3399	−5.5916	345.8367
3	244.1771	−121.0730	7.1414	178.4731	75.4911	−5.5916	345.9741
4	17.7684	270.0970	−16.0551	158.2606	109.7645	−2.2261	347.0417
5	−17.9290	267.6828	−12.0433	−153.4852	113.8857	9.4908	343.6362
6	−244.3333	−118.9524	6.6741	−174.5178	83.5386	9.5100	345.9276

Table 6
Error of position/orientation for validation poses $\mathscr{X}_v$

	Before calibration	After calibration	
		Computed with x_c	Computed with x_{c_σ}
Mean position (mm)	1.32	0.67	1.10
Std dev. position (degree)	0.62	0.4	0.45
Mean orientation (mm)	0.34	0.26	0.26
Std dev. orientation (degree)	0.27	0.27	0.27

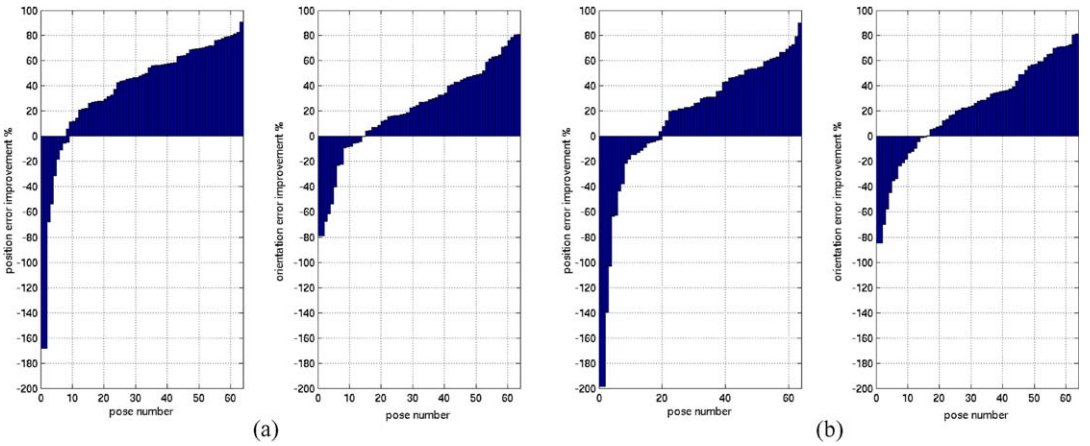

Fig. 5. Norm of the positioning error associated with each validation pose. (a) For x_c, (b) for x_{c_σ}.

results are sorted in ascending order. Table 6 and Fig. 5 show that the kinematic parameters x_c provide better results than the parameters x_{c_σ}.

4. Interval calibration

In this section, we attempt to certify the previous results using an interval approach. First we introduce the basis of interval arithmetic and interval methods, then proceed to the certification of previous experimental results.

4.1. Measurement extension to intervals

The main obstacle to proper calibration is noise in measurement devices and internal sensors, which makes the identification of kinematic parameters difficult. Two classical strategies can be used to decrease the influence of noise:

- The selection of optimal measurement configurations, whose qualities are specified to minimize the sensitivity of a parameter identification to the measurement noise.
- Taking into account the noise distribution and excluding outliers, in order to avoid or reduce the influence of *bad* measurement data.

The first point has already been accounted for in these experiments by using extremal joint positions for calibration. Consequently, this section deals now with the second strategy. As previously presented, use of the weighted least squares method requires the assumption that the error distribution is Gaussian. Even if the noise is Gaussian, however, it may be difficult to estimate the characteristics of the distribution given the complexity of the measurement procedure. As Wampler remarks in [2], another possible problem is backlash errors. These tend to be bimodal, and thus cannot even be approximated by a Gaussian distribution.

We want to relax the Gaussian hypothesis by considering each measurement datum to lie inside an interval (range of variation), without any assumptions about the type of distribution. Typically this would apply when the errors are uniformly distributed in an interval, or even more generally when the errors are only known to lie in a given range. This can be effective even if it is difficult to determine the maximal and minimal values of each measurement variable.

4.2. Certification using interval arithmetic

Methods based on interval arithmetic and interval function evaluation naturally take into account the influence of measurement noise on the calibration equations. For example, these methods allow one to assert that no kinematic parameter values in their given ranges satisfy the calibration equations associated with a set of measurements.

Interval arithmetic, introduced by Moore [25], is based on the representation of an uncertain variable x as an interval $\mathbf{x} = [\underline{x}, \bar{x}]$ reflecting a conservative worst-case estimate of the range of x. We also use the following notation related to intervals: $\inf(\mathbf{x})$ stands for $\underline{x}$, $\sup(\mathbf{x})$ for $\bar{x}$, $\mathrm{mid}(\mathbf{x})$ for $\frac{1}{2}(\underline{x} + \bar{x})$ and $\mathrm{rad}(\mathbf{x})$ for $\bar{x} - \underline{x}$.

Given a set of variables $x_1, \ldots, x_n$ lying in the corresponding intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the *interval evaluation* or interval extension of a real-valued function $f(x_1, \ldots, x_n)$ is the interval $\mathbf{f}(\mathbf{x}) \equiv [\underline{f}, \overline{f}]$ such that

$$f(x_1, \ldots, x_n) \in \mathbf{f}(\mathbf{x}), \quad \text{or equivalently} \quad \underline{f} \leqslant f(x_1, \ldots, x_n) \leqslant \overline{f}, \quad \text{for all } x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n. \tag{4}$$

In other words, $\underline{f}$ and $\overline{f}$ are bounds for the minimal and maximal values of f over the set of intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$. There are numerous ways to calculate an interval evaluation function [26] that produce reasonably good bounds for the minimum and maximum values of f. Having an estimate of $[\underline{f}, \overline{f}]$ that is as close as possible to the real minimum and maximum values of f is the key to a successful use of intervals.

The simplest method is *natural evaluation*, in which all mathematical operators in an expression for f are simply replaced with their interval equivalents. In this case, the resulting range $[\underline{f}, \overline{f}]$ is highly dependent on the symbolic expression used for f. Another interesting interval evaluation method is the *centered form* (or linear Taylor form), defined as follows:

$$\mathbf{f}_{\mathrm{T}}(\mathbf{x}) = f(x) + \mathbf{A}(\mathbf{x} - x), \tag{5}$$

where $\mathbf{A} = \mathbf{f}[x, \mathbf{x}]$ is a suitable $n \times n$ interval matrix called a slope matrix, and x is chosen arbitrarily in $\mathbf{x}$ [26].

4.3. Interval solution of calibration

The solution set of a well-determined system of equations, where the coefficients of the equations are scalars, is usually zero-dimensional. A continuous solution set results when the coefficients of a system of equations are constrained to lie within given intervals.

For an over-determined system of equations, there is generally no solution if the coefficients of the equations are scalars. We will assume that the coefficients of an over-determined set of equations are constrained to lie within known ranges, and that our objective is to determine if any solution or solutions exist. The goal is then to determine an outer approximation of the solution set, i.e. a domain that is guaranteed to include all

solutions to the system of equations. We propose to determine the solution set of the over-determined system (3) by using interval programming methods.

We assume that the uncertain coefficients M_k of Eq. (3) may take all possible values inside an interval of variation denoted by $\mathbf{M}_k$, and combine these intervals into the interval vector $\mathbf{M}$. Our goal is to determine the continuum $\mathscr{S}(\mathbf{M})$ of kinematic parameters x satisfying (3):

$$\mathscr{S}(\mathbf{M}) = \{x | f(x, M_k) = 0 \text{ with } M_k \in \mathbf{M}_k,\ k = 1, \ldots, p\}. \tag{6}$$

Determining the set $\mathscr{S}(\mathbf{M})$, which generally has a complicated shape, is a difficult problem. One can simplify it, however, by computing only a domain $\mathscr{B}$ that encloses this set. The domain $\mathscr{B}$ will include points that are not solutions of the equation system, but if the overestimation of the solution set is small, $\mathscr{B}$ will contain all relevant information about $\mathscr{S}(\mathbf{M})$.

Two possible domains for this purpose are a box enclosure and a linear enclosure (i.e. a domain whose borders are defined by a set of lines). A two-dimensional visualization of possible solution enclosures is given in Fig. 6.

In this paper, we will use a Taylor expansion to obtain a linear enclosure of $\mathscr{S}(\mathbf{M})$. Alternatively, the solution set may be obtained through the equation semantics [27]. We use linear programming methods to compute the extreme values of a linear approximation. This results in a box enclosure $\mathscr{B}$ containing $\mathscr{S}(\mathbf{M})$. Using the quadratic approximation results from [28], it is not hard to see that when the uncertainties of the M_k are on the order $O(\epsilon)$ the size of the resulting box will be at most $O(\epsilon^2)$ larger than the tightest possible box enclosing $\mathscr{S}(\mathbf{M})$. Thus, in practice the overestimation of the box enclosure has little effect on the quality of the results.

Starting from an initial estimate provided by a box solution, a fixed point algorithm is used to iteratively sharpen the solution set to a linear enclosure. The iteration ends naturally when the bounds of $\mathscr{B}$ no longer improve much, i.e. when the maximal box width does not decrease significantly in an iteration step. An even closer approximation of the solution set $\mathscr{S}(\mathbf{M})$ can be obtained by bisecting the computed box $\mathbf{x}$ and restarting the iterative process with the two resulting domains as initial estimates.

Although we tested several interval methods, we present here only the interval evaluation which provided the sharpest approximation to $\mathscr{S}(\mathbf{M})$. It is particularly well adapted to over-determined systems of equations. Since there are many more possibilities to explore, however, an improved analysis of the system is still possible.

4.3.1. Interval Newton formulation of implicit equations

Consider the vector function $F(x, M)$ with components $F_k(x, M) = f(x, M_k)$. A centered form interval extension of $F(x, M)$, performed in two steps, gives:

$$\mathbf{F}(\mathbf{x}, \mathbf{M}) = \mathbf{F}(x, \mathbf{M}) + \mathbf{A}(\mathbf{x}, \mathbf{M})(\mathbf{x} - x) = F(x, M) + \mathbf{B}(x, \mathbf{M})(\mathbf{M} - M) + \mathbf{A}(\mathbf{x}, \mathbf{M})(\mathbf{x} - x), \tag{7}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the natural interval extensions of the identification Jacobian matrices $\partial F(x, M)/\partial x$ and $\partial F(x, M)/\partial M$, respectively, which may be computed from the explicit expressions for these matrices. x and M are selected in $\mathbf{x}$ and $\mathbf{M}$ as $x = \text{mid}(\mathbf{X})$ and $M = \text{mid}(\mathbf{M})$.

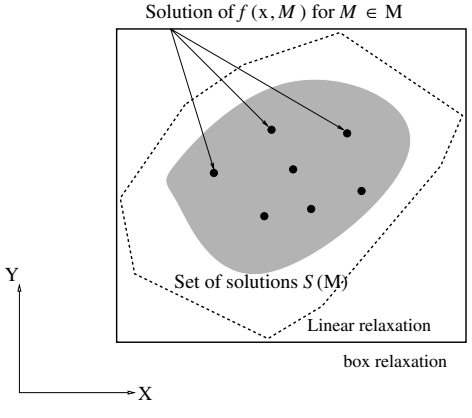

Fig. 6. 2D example of the solution set $\mathscr{S}(\mathbf{M})$.

We want to determine an enclosure $\mathscr{B}$ for the vectors x such that $F(x, M) = 0$ for some $M \in \mathbf{M}$. Given a trial enclosure $\mathscr{B}_j$ (which is guessed for $j = 0$), we want to use the information in the centered form to reduce the size of $\mathscr{B}_j$ and produce an improved enclosure $\mathscr{B}_{j+1}$. Newton's method may be extended to the interval case [26,28] through a recipe called the *Newton operator*, which is used to construct a box $N_j(x_j, \mathbf{x}_j)$ defined as an enclosure of all vectors $x \in \mathbf{x}_j$ satisfying the linear inclusion

$$A(x - x_j) \in -f(x_j, \mathbf{M}) - \mathbf{B}(x_j, \mathbf{M})(\mathbf{M} - M) \quad \text{with } A \in \mathbf{A}. \tag{8}$$

The interval Newton method is then defined by

$$\mathbf{x}_{j+1} := \mathbf{x}_j \cap N_j(\mathbf{x}_j, x_j). \tag{9}$$

The interval Newton method ends if the size of the box is no longer substantially decreased with respect to the previous iteration. This criterion is tested using the formula $\|\mathrm{rad}(\mathbf{x}_j)\|_1 - \|\mathrm{rad}(\mathbf{x}_{j+1})\|_1 < \Delta$.

There are several ways to solve for the linear inclusion (8), one of which will be presented in next section. For details on other properties (e.g. convergence, uniqueness) of the interval Newton method the reader may consult [28], in which it is shown that no solution of $F(x, M) = 0$ contained in the initial trial box $\mathbf{x}_0$ can be lost (i.e. may lie outside $\mathscr{B}_j$). As a consequence, if the intersection of $\mathscr{B}_j$ and $N_j(\mathscr{B}_j, x_j)$ is empty for some j, then since $x_{j+1} = 0$ by (9), there is no solution in the initial trial box $\mathscr{B}_0$. Moreover, if some $\mathscr{B}_{j+1}$ is strictly included in $\mathscr{B}_j$ then it is certain that $\mathscr{B}_0$ (and hence all $\mathscr{B}_j$) contains a solution of $F(x, M) = 0$ for every $M \in \mathbf{M}$. This makes the interval Newton method an excellent tool for certification computations.

4.3.2. Reformulation as a linear programming problem

As seen in the previous section, the heart of the proposed method lies in solving (8). An equivalent formulation of this problem is to find the set of solutions

$$\Sigma(\mathbf{A}, \mathbf{b}) = \{x | Ax = b, A \in \mathbf{A}, b \in \mathbf{b}\}, \tag{10}$$

where $\mathbf{A}$ is an interval matrix and $\mathbf{b}$ is an interval vector. To determine $\Sigma(\mathbf{A}, \mathbf{b})$, the tightest enclosing box, is an NP-hard problem and hence expensive in high dimension. Indeed, the shape of the set can be quite complicated. However, it is possible to find an enclosure of $\Sigma(\mathbf{A}, \mathbf{b})$ by an interval vector $\mathbf{x}$ with limited overestimation, provided that the intervals are narrow enough.

Basic interval analysis methods suitable for this problem include preconditioned Gauss elimination and Krawczyk's method (see [26,28–30]). We tested an improved algorithm proposed by Rump [31] based on these methods, and implemented it using the Matlab package INTLAB. Matlab, while highly useful for square systems of equations, is not adapted to over-determined problems. Although it can solve for them, the resulting enclosure is usually not as good as the one obtained by the method we will now describe. This method is based on a reformulation of the problem in the context of linear programming.

The method consists of two steps. In the first step, starting from a box enclosure, we overestimate $\Sigma(\mathbf{A}, \mathbf{b})$ with a convex polyhedron defined by scalar linear inequalities. In the second step, we determine the minimal and the maximal value of each component of the points in the polyhedron using linear programming methods (e.g. the simplex algorithm). This provides an enclosure $\mathbf{x}$ of $\Sigma(\mathbf{A}, \mathbf{b})$. Again, results from [28] imply that if the intervals in the entries of $\mathbf{A}$ and $\mathbf{b}$ are narrow then the overestimation will be of higher order and hence small. To improve the quality of $\mathbf{x}$, the two steps are repeated until no significant improvement is obtained.

For any matrix A_0 (which we choose as the midpoint of A) we can apply a Krawczyk-type decomposition to the problem,

$$Ax - b = (A - A_0)x - b + A_0 x \tag{11}$$

to see that any $x \in \Sigma(\mathbf{A}, \mathbf{b})$ must satisfy the linear inequalities

$$\begin{aligned} &Ux \leqslant u, \quad \text{where} \\ &U = \begin{pmatrix} A \\ -A \end{pmatrix}, \\ &u = \begin{pmatrix} -\inf((\mathbf{A} - A) \cdot \mathbf{x} - \mathbf{b}) \\ \sup((\mathbf{A} - A) \cdot \mathbf{x} - \mathbf{b}) \end{pmatrix}. \end{aligned} \tag{12}$$

This observation goes back to [32], and for narrow interval coefficients can be used to find a nearly optimal, polyhedral enclosure of $\Sigma(\mathbf{A},\mathbf{b})$. We therefore call a linear programming solver a total of $2n$ ($n = \dim(\mathbf{x})$) times to solve the problems

$$\underline{x}_k = \min\{x_k | x_k | Ux \leqslant u\}, \tag{13}$$

$$\bar{x}_k = \max\{x_k | x_k | Ux \leqslant u\} \tag{14}$$

for $k = 1,\ldots,n$. This algorithm produces an enclosing box for $\Sigma(\mathbf{A},\mathbf{b})$, and hence can be used to define the Newton operator and the interval Newton iteration discussed in the previous section.

Krawczyk-type decomposition is one possible solution to the problem of obtaining a convex over-estimation of $\Sigma(\mathbf{A},\mathbf{b})$. For this application, we also use the Beaumont decomposition [33], which is based on a theorem due to Oettli and Pragger [34]. Beaumont decomposition also produces linear inequalities, that are added to the constraints used by a linear programming algorithm to over-estimate $\Sigma(\mathbf{A},\mathbf{b})$.

5. Interval certification of experimental calibration

5.1. Certification of the least squares solutions

5.1.1. Interval extension of the experimental data

For the experimental calibration presented above, a realistic range of variation in the measurement data $M_k = [P, R, L_{1\ldots6}]_k$ can be estimated for each pose k. Each interval component of the vector $\mathbf{M}$ was given a width equal to six σ_M and centered on the mean M_0 of the corresponding measurement (i.e. $M \in \mathbf{M} = [M_0 - 3\sigma_M, M_0 + 3\sigma_M]$). If Gaussian assumptions are valid, then 99.7% of the measurement data lie inside this range.

5.1.2. Interval certification of least square solutions

We want to certify the solutions x_c and x_{c_σ} obtained in Section 3.3. For this purpose, two questions need to be answered:

- *Do the least squares solutions vanish the calibration equations for some realization of the measurement variables within the stated intervals?*
  To answer this, we evaluated (3) in both x_c and x_{c_σ} using interval arithmetic for each of the measurements $M_k \in \mathbf{M}_k$, $k = 1,\ldots,N$. We have observed that the null vector $\mathbf{0}$ belongs to neither $f(x_c,\mathbf{M}_k)$ nor $f(x_{c_\sigma},\mathbf{M}_k)$ for any $M_k \in \mathbf{M}_k$. In other words, there are no measurements $[P, R, L]_k$, chosen inside an interval $\mathbf{M}_k$, such that the system of calibration equations is canceled for kinematic parameters equal to x_c or x_{c_σ}. In this sense, these solutions are not certified.
- *Does a solution to the calibration equations exist for all possible realizations of the measurement variables within the stated intervals?*
  The answer is *negative*, since the algorithm described in Section 4.3 cannot be used to find a set of scalar values for the kinematic parameters x such that the calibration system of equations is canceled for $M_k \in \mathbf{M}_k$. Indeed, we can certify that there is no solution for $\mathbf{x}_\Delta \in [x_c - \Delta, x_c + \Delta]$ with $\Delta = 100$ mm (i.e. $\mathbf{0} \notin f(\mathbf{x}_\Delta,\mathbf{M}_k)$).

5.1.3. Discussion

Two hypotheses may explain why there is no solution to the interval equations $f(x,\mathbf{M}_k) = 0$:

H-1 the standard deviation of the measurement noise is under-estimated; in this case, the width of the intervals $\mathbf{M}_k$ must be enlarged.

H-2 the equation $f(x,\mathbf{M}_k) = 0$ does not reflect the actual robot model. In other words, the simplifying assumptions leading to the kinematic equations are not valid.

The first hypothesis is highly improbable. Indeed, since the width of the measurement interval $\mathbf{M}_k$ is chosen to be three times the standard deviation, there is little chance that the variation of the measurement data is under-estimated for such a large number of measurements. On the other hand, two essential assumptions of our robot model are that the articulations are perfect and perfectly assembled, and that there is no joint backlash. Hence, hypothesis H-2 is the more probable and we propose a small modification of the calibration equations in the next section.

5.2. Shifted equation

A simple way to take into account the difference between the model and the physical system is by adding uncertainties to the calibration equations. For this purpose, we will add a variable $_i$ for each of the six legs such that

$$g(x_i, M_{i,k}) = \|P_k + R_k \cdot b_i - a_i\| - (L_{i,k} + l_i) = \Upsilon \quad \text{with } x_i = [a_i, b_i, l_i] \text{ and } M_{i,k} = [P_k, R_k, L_{i,k}]. \tag{15}$$

Substituting the interval $[-\epsilon, +\epsilon]$ for allows one to incorporate errors such as joint imperfection and backlash into the kinematic model.

We now determine the minimum value of ϵ, as a function of the measurement error given by the interval vector $\mathbf{M}_k$, such that

$$\{x | g(x, M_k) = [-\epsilon, +\epsilon] \text{ with } M_k \in \mathbf{M}_k,\ k = 1, \ldots, N\} \neq \emptyset. \tag{16}$$

In other words, we will determine the minimum value of ϵ required to allow the calibration equations at least one solution. If this value is large, then it will be necessary to reconsider the kinematic model.

5.2.1. Solution search algorithm

An algorithm we designed optimizes the value of ϵ associated with each leg, and finds a *valid point* corresponding to this value of ϵ. A valid point is a vector of scalar values x_0 such that $g(x_0, M_k) \cap [-\epsilon, +\epsilon] \neq \emptyset$.

We use a simple bisection procedure based on the following fact: if there exists a valid point for a certain ϵ, it is highly probable that an infinite number of points are also valid for all ϵ' greater than ϵ. We therefore cannot optimize ϵ by launching the solver and waiting for an answer.

Assume that we have starting guesses for lower and upper bounds on ϵ, which we denote ϵ_- and ϵ_+ respectively. The optimum value is found by repeating the following operations in a script:

- Ask the solver to find a valid point in $[x_c - \Delta, x_c + \Delta]$, in less time than a specified number of seconds (solving is based on the method described in Section 4.3).
- If no solution is found, set ϵ_- to the current value of ϵ, and increase ϵ closer to ϵ_+.
- If the solver times out, there might be a continuum of solutions. In this case decrease ϵ (so it is closer to ϵ_-). We cannot affect ϵ_+, since the solver was interrupted. If the distance between ϵ and ϵ_- is too small, it means that we cannot isolate a solution within the specified delay. In this case we double the delay and set the value of ϵ back to the average of ϵ_- and ϵ_+.

When the process ends, it returns a small box $x \pm w$ as well as an estimated error ϵ_{cert}. In order to certify a solution, we pick a point inside this box and adjust the value of ϵ until all the constraints are satisfied. The resulting value, ϵ, is usually only a few percent greater than the one given by the script above.

Before running the script, ϵ_+ can be initialized by applying the previous method to the least squares solution. ϵ_- can be set to 0.

5.2.2. Results

Execution of the previous algorithm demonstrates that for each leg $i = 1, \ldots, 6$, the uncertainty of the model ϵ is necessarily greater than the value ϵ_{cert} given in Table 7. Below this value, our algorithm certifies that no solutions x_i exist in the range $[x_c - \Delta, x_c + \Delta]$ with $\Delta = 100$ mm (including x_c).

Table 7
Minimal certified values of ϵ_{cert} in (mm)

Leg's number	1	2	3	4	5	6
ϵ_{cert} (mm)	27.38	26.93	32.78	33.42	25.46	24.01
ϵ (mm)	27.53	26.96	33.12	33.48	25.53	24.24

Because the values found for ϵ_{cert} are large with respect to the length of the legs, we can conclude that the kinematic model used to derive the calibration equations is oversimplified.

6. Conclusion

This paper presented an experimental validation of the interval calibration method. This original technique is based on representation of the data as intervals rather than discrete variables. It therefore incorporates the uncertainties associated with measurement variables into the calibration model problem directly.

The first step of the approach is to determine a realistic range in which the measurement variables are nearly always located. It is often possible to easily obtain this range by evaluating the sensor noise associated with the measurements, which is much more realistic than assuming a Gaussian error distribution. The method is validated through an accurate and rigourous experimental process. Vision-based experiments yield the intervals associated with the position and orientation measurements, which automatically include both measurement noise and robot repeatability error.

The main contribution of this paper is the introduction of another approach to the calibration problem. Instead of finding the kinematic parameter values that provide the best fit to the calibration equations (while not ensuring that these equations are satisfied), as is the case for the least squares approach, the interval method finds ranges for the parameters for which the calibration equations are *effectively* satisfied.

Based on measurements obtained for an experimental robot, our technique shows that there is no solution to the calibration problem that cancels the system of constraint equations even if uncertainties are introduced into the kinematic equations. As this result is certified for all possible values of the measurement data within a realistic range, we can deduce that the assumptions used to establish the kinematic model are too strong. Effects such as joint imperfections and backslashes that are not considered in the simplified calibration model cannot be neglected after all.

An optimization algorithm was also demonstrated that calculates for each leg the minimum level of additional uncertainty that must be introduced into the calibration equations to obtain a possible solution. Such information is very important to the quantification and certification of the kinematic model used to establish the calibration equations.

Future work associated with the interval technique concerns improvements in the model correction. By shifting the value of ϵ to produce a certified solution, we found a zeroth-order approximation to the error resulting from abusive simplification of the inverse kinematic model. We plan to study first-order (with respect to the measurement variables) and even higher order approximations of this error, to see if the results can be improved.

Acknowledgents

The authors would like to warmly thank Arnold Neumaier from the University of Wien for his interesting discussions, and Jean-Marc Lavest from LASMEA for providing them with the vision software described in [18].

This work was supported by the CNRS ROBEA MP2 and the Région d'Auvergne AuverFiabilis projects.

References

[1] D. Daney, Kinematic calibration of the Gough platform, Robotica 21 (6) (2003) 677–690.

[2] C.W. Wampler, J.M. Hollerbach, T. Arai, An implicit loop method for kinematic calibration and its application to closed-chain mechanisms, IEEE Trans. Robot. Autom. 11 (5) (1995) 710–724.

[3] D. Daney, Y. Papegay, A. Neumaier, Interval methods for certification of the kinematic calibration of parallel robots, in: IEEE ICRA, April–May 2004, pp. 147–152.
[4] Z. Geng, L.S. Haynes, An effective kinematics calibration method for Stewart platform, in: ISRAM, Hawäi, 15–17 August 1994, pp. 87–92.
[5] H. Zhuang, J. Yan, O. Masory, Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals, J. Robot. Syst. 15 (7) (1998) 395–405.
[6] C. Innocenti, Algorithms for kinematic calibration of fully-parallel manipulators, in: B. Ravani, J.-P. Merlet (Eds.), Computational Kinematics, Kluwer, 1995, pp. 241–250.
[7] D. Daney, Optimal measurement configurations for Gough platform calibration, in: IEEE ICRA, May 2002, pp. 147–152.
[8] A. Nahvi, J.M. Hollerbach, The noise amplification index for optimal pose selection in robot calibration, in: IEEE ICRA, 1996, pp. 647–654.
[9] H. Zhuang, Z.S. Roth, Camera-Aided Robot Calibration, CRC Press, 1996, ISBN 0-8493-9407-4.
[10] J.M.S.T. Motta, G.C. de Carvalho, R.S. McMaster, Robot calibration using a 3d vision-based measurement system with a single camera, Robot. Comput. Integrat. Manuf. (17) (2001) 487–497.
[11] H. Zou, L. Notash, Discussion on the camera-aided calibration of parallel manipulators, in: Proc. 2001 CCToMM Symp. Mechanisms, Machines, and Mechatronic, Saint-Hubert, 2001.
[12] N. Andreff, P. Renaud, P. Martinet, F. Pierrot, Vision-based kinematic calibration of an H4 parallel mechanism: practical accuracies, Ind. Robot. Int. J. 31 (3) (2004) 273–283.
[13] V.E. Gough, S.G. Whitehall, Universal tyre test machine, in: Proc. FISITA 9th Int. Tech. Cong., May 1962, pp. 117–137.
[14] D. Stewart, A platform with six degrees of freedom, in: Proc. IMechE (London), vol. 180, 1965, pp. 371–386.
[15] P. Vischer, R. Clavel, Kinematic calibration of the parallel Delta robot, Robotica 16 (2) (1998) 207–218.
[16] O. Masory, J. Wang, H. Zhuang, Kinematic modeling and calibration of a Stewart platform, Adv. Robot. 11 (1997) 519–539.
[17] M.L. Husty, An algorithm for solving the direct kinematic of Stewart–Gough-type platforms, Mech. Mach. Theory 31 (4) (1996) 365–380.
[18] J.M. Lavest, M. Viala, M. Dhome, Do we really need an accurate calibration pattern to acheive a reliable camera calibration, in: Euro. Conf. Computer Vision (ECCV'98), Freilburg, Germany, 1998, pp. 158–174.
[19] P. Brand, R. Mohr, Accuracy in image measure, in: SPIE Conf. Videometrics III, Boston, MA, vol. 2350, 1994, pp. 218–228.
[20] O. Faugeras, Three-Dimensional Computer Vision – A Geometric Viewpoint, Artificial Intelligence, Cambridge, MA, 1993.
[21] R.Y. Tsai, R.K. Lenz, A new technique for fully autonomous and efficient 3d robotics hand/eye calibration, IEEE Trans. Robot. Autom. 5 (3) (1989) 345–358.
[22] R. Horaud, F. Dornaika, Hand-eye calibration, Int. J. Robot. Res. 14 (3) (1995) 195–210.
[23] K. Daniilidis, E. Bayro-corrochano, The dual quaternion approach to hand-eye calibration, in: Proc. IAPR Int. Conf. Pattern Recognition, 1996, pp. 318–322.
[24] N. Andreff, R. Horaud, B. Espiau, Robot hand-eye calibration using structure from motion, Int. J. Robot. Res. 20 (3) (2001) 228–248.
[25] R. Moore, Interval Analysis, Prentice Hall, 1996.
[26] E. Hansen (Ed.), Global Optimization using Interval Analysis, Marcel Dekker, 1992.
[27] Y. Lebbah, M. Rueher, C. Michel, A global filtering algorithm for handling systems of quadratic equations and inequations, Lect. Notes Comput. Sci. 2470 (2002) 109–123.
[28] A. Neumaier, Interval Methods for Systems of EquationsEncylopedia of Mathematics and its Applications, vol. 37, Cambridge University Press, Cambridge, UK, 1990.
[29] R. Krawczyk, Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, Computing 4 (1969) 187–201.
[30] A. Neumaier, Introduction to Numerical Analysis, Cambridge Univ. Press, Cambridge, 2001.
[31] S.M. Rump, Solving algebraic systems with high accuracy, in: U. Kulisch, W. Miranker (Eds.), A New Approach to Scientific Computation, Academic Press, San Diego, 1983, pp. 51–120.
[32] A. Neumaier, The enclosure of solutions of parameter-dependent systems of equations, in: R.E. Moore (Ed.), Reliability in Computing, Academic Press, 1988, pp. 269–286.
[33] O. Beaumont, Algorithmique pour les intervalles: Comment obtenir um résultat sûr quandles données sont incertaines, Phd, Université de Rennes 1, Rennes, November 1999.
[34] W. Oettli, W. Prager, Computability of approximate solution of linear equations with given error bounds for coefficients and right-hand sides, Numer. Math. 6 (1964) 405–409.