Variable elimination for reliable parallel robot calibration

DAVID DANEY, IOANNIS Z. EMIRIS

INRIA Sophia-Antipolis, B.P. 93, 06902 France
{David.Daney,Ioannis.Emiris}@inria.fr

Abstract: *We study overconstrained polynomial systems from an algebraic and numerical perspective in order to improve the accuracy and reliability of Gough platform calibration. By elimination theory, we may limit ourselves to measuring only the position information by external sensors, besides internal sensor measurements. Measuring orientation, which is intricate and error-prone, becomes unnecessary. A planar analogue is used to formally develop the technique and analyze observability by means of the Jacobian. Several experiments are used to compare the existing alternatives with partial information to our approach, by numerical simulation. It is shown that our method significantly improves robustness without compromising accuracy, especially when the measured configurations lie at the workspace's boundary.*

Keywords: parallel robot, robust calibration, algebraic elimination, boundary configurations, optimization.

1 Introduction

Parallel robots, in particular the Gough/Stewart platform, offers high accuracy for manipulating heavy objects, even at large velocities. Hence its use in an array of industrial applications, from flight simulation to high-speed machine-tools lately. In the other end of the spectrum, parallel robots are entering the realm of medical applications such as surgery.

The geometry of robotic manipulators does not exactly match the design goals because of manufacturing or assembly defects. As a result, the manipulator may be plagued by reduced accuracy, since its control strongly depends on the precision of the kinematic model. One way to handle this problem is by improving the kinematic model via kinematic calibration. Let us employ the paradigm of [29] in stating a unified calibration formulation. First, given the unknown kinematic parameters $\mathcal{P}$ and the measurements $\mathcal{M}$ provided by sensors, certain loop equations $f(\mathcal{P}, \mathcal{M}) \simeq 0$ can be determined; the equations vanish within the measurement error. These constraints supply redundant information on the state of the robot.

The simplest way to determine $\mathcal{M}$ is by the internal sensors used to control the manipulator. Usually, though, their cardinality is minimal for controlling the manipulator's degrees of freedom (dof), so they do not provide redundant information. It is possible to install additional captors on passive articulations for self-calibration [30] (and simplify the forward kinematics [1]). Alternatively, one can decrease the dof by constraining the end-effector [2, 11] or the mobility of some articulation [18, 10]; this can also be used to simplify the calibration problem in the presence of external measurements [32, 15, 17].

In practice applications, it is not easy to add redundant sensors or constraints. The majority of calibration methods use external measurements to get the required information, usually by measurement devices, such as laser trackers, theodolites [22], cameras, or mechanical devices. $\mathcal{P}$ and $\mathcal{M}$ will be related by some kinematic model. For a sufficient number of measurements, the loop equations can be solved by optimization [21], linearization [15] or direct system solving [17, 28].

The basic calibration methods with external measurements use either the forward kinematics model (direct method) [21, 27] or the inverse kinematics model (inverse method) [31]. Each method has advantages and drawbacks: The latter is more robust because it decouples the problem into the calibration of each leg independently, and because the constraints admit an analytic form. The direct method faces more serious stability issues since it relies on a numerical solution for the forward kinematics, with problematic convergence in the presence of noise. On the other hand, it needs only partial measurements of the end-effector's pose. For instance, [3] avoids the position measurements and uses inclinometers to obtain the rotation parameters; on the downside, a careful choice of measured configurations is crucial for controlling the convergence.

The major contribution of this paper is to combine the advantages of direct calibration with the robustness of a new algorithm which requires only part of the external measures. Our idea is to eliminate the rotation parameters because it is easier and faster in practice to measure accurately the position. This is motivated by our experience with machine-tool construction based on parallel robots at an industrial scale (e.g., by CMW, Constructions Mécaniques de Vosges, France). Variable elimination has led to certain important breakthroughs in diverse areas, including robotics and vision [16, 25, 26]. It yields an equivalent, yet simpler, system, on which we apply an array of optimization techniques. Preliminary results were reported in [12, 13].

The next section describes Gough platform kinematics and existing calibration approaches. Section 3 sketches our algorithm and presents variable elimination for the planar and spatial cases. Section 3.4 concerns our computational tools. Simulation results for varying error and different initial estimates show, in section 4, that our method significantly enhances reliability. Different choices for the measured configurations are discussed in order to illustrate the relevance of picking them near the workspace's boundary.

2 Kinematics and Calibration

This section describes kinematics models and calibration approaches; for previous work, see also section 1. The manipulator consists of two rigid bodies, the base and the mobile platform, connected by 6 legs; see figure 1. The legs' linear actuators provide 6 dof for the platform's pose relative to the base, corresponding to position P and rotation matrix R. A configuration $\mathcal{X} = [P, R]$ is associated to 6 length variations L_i measured by "internal" sensors, $1 \leq i \leq 6$. Each leg is attached to the base by a U-joint and to the platform by a ball joint, so 23 parameters are required to model each leg [28]. But as shown in [21], the principal source of error in positioning is due to limited knowledge of the joints' centers and that part of the length not given by the sensors. We thus use a simpler model with attachment points a_i in the base frame, b_i in the mobile frame and offset lengths l_i. This gives 7 parameters per leg, therefore 42 overall, denoted

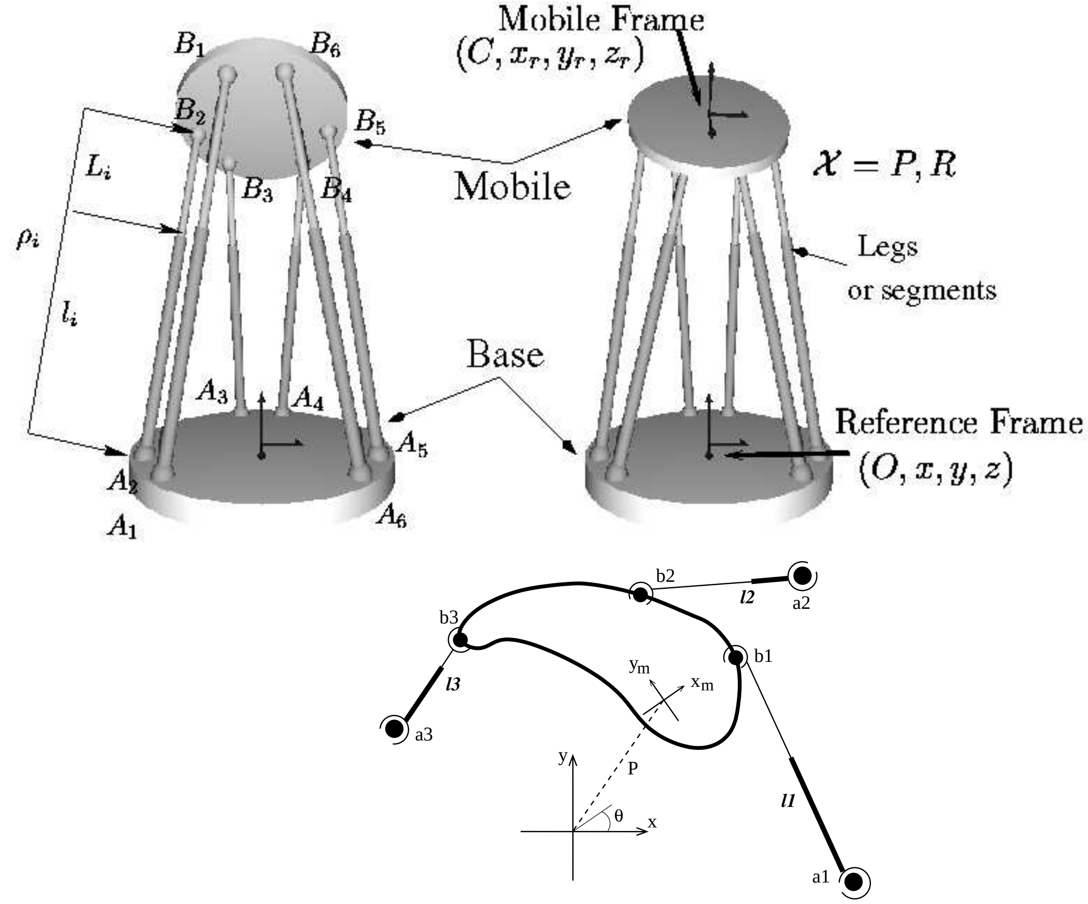

Figure 1: Gough platform and planar analogue.

by $\mathcal{P}$. The *planar robot* is studied in order to better understand the behaviour of our method by examining the constraints formally. It has 3 prismatic joints and 15 kinematic parameters (figure 1).

Kinematics models are essential for controlling the robots, and yield the constraints used in calibration. *Inverse kinematics* express the length for the i-th leg as a function of $\mathcal{X}$:

$$\|P + R.b_i - a_i\|_2^2 = (L_i + l_i)^2,\ 1 \leq i \leq 6. \tag{1}$$

To describe the rotation matrix, we use normalized quaternions with three independent parameters q_1, q_2, q_3, thus producing algebraic equations. In the planar case, there is a single parameter q. For the Gough platform, the *forward kinematics* model is more difficult to compute since it consists in solving (1) for P and R given L_i and $\mathcal{P}$. In general there are up to 40 complex solutions. A complete and exact solution (e.g. [16]) is today too costly to be useful for calibration, so we may use a Newton-Raphson scheme, e.g. [23].

The aim of calibration is to reduce the positioning error of the end-effector by accurately identifying the kinematic parameters. The *direct method* emanates from serial robot calibration. It compares the true pose of the end-effector $\mathcal{X}^m$, provided by measurement, to the estimated pose $\mathcal{X}^F$, provided by forward kinematics. The displacement error $\mathcal{C}_{\text{dir}}$ is their difference and is a function of the measurements and of parameters $\mathcal{P}$. To find the parameters that minimize the error, we need at least as many independent constraints as the number of parameters. Let N be the number of measured configurations; for each, the number of constraints k depends on the dimension of the measured vector $\mathcal{X}^m$. To identify the 42 kinematic parameters when some device measures all 6 dof (hence $k = 6$), we must have $6N \geq 42$. Typically, nonlinear optimization is applied with N measures to determine the parameters minimizing $\sum_{t=1}^{N} \|\mathcal{C}_{\text{dir}}^t(\mathcal{X}_t^m, L_{t=1,\ldots,6}, \mathcal{P})\|^2$ in a least-squares sense.

The *inverse method* presents a more popular alternative by quantifying the pose error indirectly through the leg length error. Each constraint follows from (1) and depends only on the parameters of the respective leg, so we can decouple the robot calibration in 6 subproblems and take $N \geq 7$ measures. To solve these equations in terms of the parameters, one typically uses optimization (the analytic Jacobian is given in [31]), but it is also possible to use dyalitic elimination [17, 11] or linearization [28, 11]. The downside is that this technique requires knowledge of the position and orientation of each measured configuration.

Concerning the choice of measured configurations, the inverse method behaves more robustly near the workspace's boundary [11]. This is why we use both random and boundary configurations; see section 4.

3 Calibration with Partial Measurements

Our calibration method seeks to combine the advantages of the two basic methods: we can dispense of part of the pose information on the platform, while achieving higher numerical stability and, therefore, better robustness in regard to measurement errors, than the direct method. The reason is that we manage to express analytically the constraints. Our basic tool is variable elimination, in other words deriving an equivalent system to (1) which does not include the q_i's. We could have eliminated some geometric parameters, thus obtaining a system with fewer unknowns to solve. Yet, in practice, it is more useful to eliminate some measurement information in order to simplify the actual process of calibration. We chose to eliminate the 3 (1 if planar) rotation parameters because they are harder to measure and introduce more noise.

3.1 Algebraic Elimination

Our approach combines symbolic variable elimination with numeric optimization. We focus on resultants and their extensions since they respond precisely to the task of eliminating the rotation parameter(s); see (3). The *resultant* (or *eliminant*) is a single polynomial in m variables that vanishes if and only if there exists some common zero among $n + 1$ equations in $n + m$ variables. In the spatial (resp. planar) case $n = 3$ (resp. 1), m equals the number of unknown parameters. The resultant will be expressed either as a determinant or a nontrivial divisor of this determinant; see [5, 6, 14, 25, 26] and their references.

The resultant is a necessary and sufficient condition, but sometimes its degree causes computation to slow down. We can construct smaller conditions, which are just necessary: Any *square* matrix with columns (resp. rows) corresponding to powers of the eliminated variables (resp. multiples of the input polynomials) yields a determinant that vanishes whenever the system has a common root. This is known as *dialytic elimination* which essentially considers the system as a linear one with the new variables being the monomials. The C program of [14] constructs such matrices in an incremental fashion.

The resultant captures the solvability of $n+1$ polynomials in n variables. Yet, the systems at hand contain more than $n+1$ polynomials, yielding a substantially harder problem. For overdetermined systems, solvability conditions have been much less studied; one starting point may be [5, 9, 19, 20]. In particular, the size of the matrices that we need to consider is not known. There are two reasons for which using only necessary conditions yields accurate solutions to our problems. First, the systems we treat have at least one solution, which corresponds to the physical system; moreover, this solution is generically unique. Second, using several instances of the constraints yields, through optimization, this solution.

3.2 The Case of Planar Robots

This case is studied as a testbed of our ideas, since we can better analyze the algebraic and numerical behaviour of our tools on the various formulations of the constraints. The main motivation is thus to look carefully into the side-effects of elimination. On the other hand, the possibility to solve analytically the forward kinematics of the planar robot implies that the direct method above offers an accurate approach that does not rely on algebraic elimination [23].

We use a planar analogue of the Gough platform. The mobile part is connected to the base by 3 prismatic joints. Equations (1) become

$$F_i = \left\| P + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} b_i - a_i \right\|_2^2 - (L_i + l_i)^2, \qquad 1 \le i \le 3, \tag{2}$$

for each measured configuration. These may be written $F_i = c_{i0} + c_{i1} q + c_{i2} q^2$, where $q = \tan(\theta/2)$ and the c_{it} depend on the measurements and the unknowns but not on q; figure 1. For 2 univariate polynomials the resultant equals the Sylvester determinant, with rows indexed by multiples of the polynomials and columns indexed by powers of the eliminated variable. The Sylvester matrix of F_i, F_j is

$$\forall i,j \in \{1,2,3\}, \qquad \begin{array}{c} \\ F_i \\ qF_i \\ F_j \\ qF_j \end{array} \quad S = \begin{array}{c} \begin{array}{cccc} 1 & q & q^2 & q^3 \end{array} \\ \begin{bmatrix} c_{i0} & c_{i1} & c_{i2} & 0 \\ 0 & c_{i0} & c_{i1} & c_{i2} \\ c_{j0} & c_{j1} & c_{j2} & 0 \\ 0 & c_{j0} & c_{j1} & c_{j2} \end{bmatrix} \end{array} \Rightarrow \deg\det S(c) = 4.$$

These determinants provide a set of necessary equations, denoted by $G_j(P, L_i, \mathcal{P}) \simeq 0$; there are $\binom{3}{2} = 3$ possible G_j's. Let k stand for a lower bound on the number of algebraically independent G_j's, with $k = 2$ (unless the c_{ij} are in a singular position). This system represents case 2.2 in table 1. We may improve numerical stability by using all 3 constraints (case 2.3).

In the experiments, the high degree of the resultant was an important obstacle to speed and accuracy. But we can find smaller necessary conditions by dialytic elimination. Such is the 3×3 matrix with rows F_1, F_2, F_3 and columns $1, q, q^2$, yielding a constraint of degree 3 (case 1.1).

Here $\nu = 3$ hence sufficiency is only guaranteed by the rank drop of a 6×4 matrix. It is easy to show that this is equivalent to the vanishing of all 3 resultants and the last 3×3 determinant. Hence all 3 cases here express necessary conditions. They are all generalized to the spatial robot.

Before solving for the 15 unknown parameters, we should decide whether they are observable. For this, let us construct the Jacobian of the G_j instantiated at as many measured configurations (or more) as there are parameters. The matrix can be computed formally, but we observe that it *drops rank* by one! This behavior is repeated in the spatial case.

Physically, eliminating the rotation variable has removed any point of reference for the orientation of the mobile frame. More formally, recall that the resultant is a condition of root existence over an algebraically closed field, in our case this being the complexes. So while eliminating q, we have no control on its taking complex values. Consider the planar parametrization of b_i using distances and relative angles: $b_1 = \frac{d_1}{1+t_1^2}(1-t_1^2, 2t_1)$, where t_1 is the tangent of a half-angle, also used to parametrize b_2, b_3. Then every resultant contains factor $(1+t_1^2)^4$. In fact, $F_i = (1+t_1^2)Q_i + 4d_iP_i(q+\sqrt{-1})^2$, where $1 \leq i \leq 3$, for Q_i, P_i independent of q, t_1. A manifold of spurious solutions corresponds to $q = -\sqrt{-1}, t_1 = \sqrt{-1}$ and causes the Jacobian to drop rank by 1.

Such facts were used in [4] in order to automate the determination of observable parameters. They apply a QR factorization of the Jacobian matrix in order to identify any possible relations between parameters. The direct method is used for calibration with partial information as in the present article and their results confirm ours.

Here, relying on necessary conditions is good enough, as in the spatial case, because the system at hand has a unique solution, which is indeed approximated, since we employ several instances of the constraints.

Table 1: Elimination cases in one dimension.

Case	Number of equations per elimination	k / number of constraints	$\deg G_j$	N
1.1	3	1/1	3	14
2.2	2	2/2	4	7
2.3	2	2/3	4	7

To exclude the extraneous solutions, we must fix as many parameters as the eliminated variables. Then, the Jacobian has full rank as we confirmed formally in the planar case and numerically in the spatial case. If $(\mathbf{x_m}, \mathbf{y_m})$ represents the mobile frame, we set b_1 on the $\mathbf{y_m}$-axis so there are 14 parameters to compute. In table 1, N stands for the minimum number of distinct configurations required for estimating 14 parameters in the various cases.

3.3 The Case of Spatial Robots

Analogously to the planar case, eliminating q_1, q_2, q_3 leads to a rank-deficient Jacobian. For all parameters to be observable, we specify a reference frame for the orientation by fixing 3 parameters. Let $(C, \mathbf{x_m}, \mathbf{y_m}, \mathbf{z_m})$ represent the mobile frame, then set b_1 on the $\mathbf{x_m}$-axis and b_2 on the $\mathbf{x_m}$-$\mathbf{y_m}$ plane. The minimal number of parameters is thus 39 and this does not complicate the calibration process, since we seek the relative position of the points. The results in [4] confirm this number (39) as the minimal number of observable parameters.

System (1) defines 6 polynomials F_i with R parametrized by q_1, q_2, q_3. Eliminating them yields at least k independent G_i's for each measured configuration:

$$F_{1,\ldots,6}(P, R, L_i, \mathcal{P}) \to G_{1,\ldots,k}(P, L_i, \mathcal{P}), \qquad 1 \leq k. \tag{3}$$

Since any subset of 4 polynomials defines a resultant, there are $\binom{6}{4} = 15$ possible resultants. Different matrices by the methods of Macaulay and the sparse resultant [6, 14], or Dixon and Bézout [25], yield determinants divisible by the resultant. We have experimented mostly with an optimal matrix from [25] which is of hybrid (or Dixon) type: 4 rows express the Bézoutian, and another 16 are constant or linear multiples with each of the 4 polynomials. The resultant degree is 32; this is case 3.3 in table 2. The high degree of the resultant is again an obstacle in optimization since it slows down convergence.

We look for square matrices whose rows correspond to multiples of the input polynomials. As in the planar case, dialytic elimination defines necessary conditions by taking 5 polynomials, each multiplied by $1, q_1, q_2, q_3$. The columns are indexed by power products $1, q_1, q_2, q_3, q_1q_2, q_2q_3, q_3q_1, q_1^3, q_2^3, q_3^3,\ q_1^2q_2, q_1^2q_3, q_2^2q_1, q_2^2q_3, q_3^2q_1, q_3^2q_2$, and the matrix is indeed square, of dimension 20. This yields $\binom{6}{5} = 6$ constraints. Subcase 2.2 considers the minimum number of such constraints, namely 2. Subcase 2.6 uses all 6 constraints produced.

Another possibility is to use all 6 polynomials, each multiplied by $1, q_i, q_j$, and another 2 polynomials multiplied by q_t, for $i, j, t \in \{1, 2, 3\}$, yielding a 20×20 matrix, with the same column monomials as in case 2. This is case 1.1, producing a unique constraint of degree 20.

Note that $\nu = 5$ calls for a 120×56 matrix, which would have a very high degree. The smaller matrices we use give accurate results (section 4) and validate our experimental approach. As above, this is explained by the uniqueness of solution and the high number of constraint instances.

To determine 39 parameters, we use N measurements with k constraints such that $kN \geq 39$. With the classical direct method $k = 3$ so $N \geq 13$ measurements. Every constraint is instantiated at N measurements, denoted by G_j^t, where $1 \leq t \leq N$. All $G_j^t \simeq 0$ at the true solution.

Table 2: Elimination cases in 3 dimensions.

case	Number of equations per elimination	k / number of constraints	$\deg G_j$	N
1.1	6	1/1	20	39
2.2 & 2.6	5	2/6	20	20
3.3(resultant)	4	3/15	32	13

3.4 Solving for the Geometric Parameters

The problem, in both planar and spatial cases, is now reduced to solving the G_j^t's, for $1 \leq t \leq N$, which are independent of any rotation variable. For the planar case, the analytic expression of the determinants G_j (and the respective Jacobians) have been computed on MAPLE. The spatial constraints are too large, however. This leads us to consider a numerical evaluation of the corresponding determinants in MATLAB, whereas the Jacobian can be obtained by finite differencing.

Different methods can be applied to solve the constraints G_j. We present our numerical results which concern a standard approach based on nonlinear least-squares minimization of $\sum_{t=1}^{N} \sum_j \|G_j^t(\mathcal{X}_t^m, L_{t=1,\ldots,6}, \mathcal{P})\|^2$. For this, we used the optimization toolbox of MATLAB 5.3 and command `lsqnonlin`, which chooses the Large-Scale Method [7]. To avoid explicit evaluation of the determinants, the criterion consists in minimizing the smallest matrix singular value, obtained by singular valued decomposition (SVD). We have also used the ratio of the smallest to the largest singular value, with equally good results. On the other hand, using the ratio of two successive singular values, even if they are the two smallest values, does not yield as good a convergence. The next section includes a discussion on computational complexity.

4 Simulation Results

We start with the geometric parameters of an existing robot; to be precise the left-hand robot of INRIA, e.g. [23]. These values, given by the constructor, serve as initial estimates of the sought parameters. They are perturbed by at most 1mm or 0.1mm, and denoted $\mathcal{P}^r$, to simulate a realistic robot design. These parameters serve to solve (1) for L_i using randomly generated poses. When measurement quantities are exact, all methods calibrate the robot fast and accurately, except from method 3.3 which suffers from the high degree of the constraints. Below we concentrate on noisy measurements, with error uniformly distributed either in $[0, 5]\mu$m or in $[0, 10]\mu$m. This is well above standard errors by the devices used in practice, which usually have errors in the order of 10μm or even down to 1μm.

We have experimented with all elimination strategies discussed in section 3 on the spatial robot. MATLAB converges after 5 to 8 iterations per method, except for case 3.3 which is slower and gives consistently the poorest results, hence is not included in the figures below. This behavior is due to the high degree of the resultant. Experiments with all methods for the planar robot confirm our conclusions, though convergence is faster and more accurate since we formally compute each constraint and its derivatives.

Each figure shows, on the vertical axis, the error in the parameters in cm. The first column corresponds to the situation before calibration and gives the norm of the difference vector (or scalar for the l_i) between the constructor's values and $\mathcal{P}^r$. The following columns show the difference norm between $\mathcal{P}^r$ and the values obtained by one of our methods of table 2; when we compare with the classical direct method, the latter appears in the last column.

Figure 2 reports on the corresponding experiments with the planar robot, just for the sake of completeness, since the forward kinematics here admit a fast and robust solution. Our method is very accurate, even with large initial and measurement errors.

Figure 3 shows that case 1.1 gives mediocre results because it requires a very large set of measurements. Since it uses a single constraint, it converges very fast and could serve in a more general context than robot calibration, when accuracy is less of a concern than time. When elimination yields more independent constraints, we obtain the best results, especially in case 2.6, where the use of 6 constraints helps improve numerical accuracy. The direct method suffers from the lack of an analytic form for the constraints, which makes it employ an extra optimization loop.

In the rest of the section we focus on case 2.6, which performs best. The 3 experiments in figure 4 compare it (middle column) with the classical direct method (last column). Figure 4(a) reflects the effects of varying the number of measured configurations up to 80, when all other characteristics of the experiment stay the same. It is intuitive that more measured configurations lead to better results with out algorithm, at higher computational cost, of course. This improvement is limited after a certain number of measurements [11].

Concerning initial error, we have used (but not shown) errors up to 10mm: they exhibit fast convergence but do not represent realistic situations. Figure 4(b) considers estimates by the constructor which are accurate to 0.1mm. Our algorithm is comparable to the direct method and has positive gains for somewhat more than half of the parameters. Measurement error is again bounded by 0.005mm, which is crucial if we wish to be able to have these positive gains. This is realistic based on our experience with industrial practices at CMW. A feature of our algorithm is its roughly constant execution time, whereas the time complexity of the direct method increases significantly in order to achieve convergence.

Figure 4(c) illustrates the effect of reducing the measurement errors down to 0.01mm, holding other characteristics constant. Our algorithm has a net positive gain for every parameter, whereas the classical direct method performs clearly more poorly.

Figure 5 compares random configurations against those at the workspace's boundary i.e., at minimal and maximal leg lengths. It is shown that our algorithm gives excellent results at boundary configurations. This is in contrast to the direct method, which has difficulty in converging. Hence, a critical feature of our algorithm is its reliability regardless of the configurations chosen.

Lastly, we report on practical complexity concerning evaluation of the constraints by means of least-square optimization. This requires an SVD computation per constraint, with execution time depending on the matrix dimension and its determinant's degree. Recall that the elimination matrix is computed off-line in order to provide a general matrix, having as parameters the position measurements and the unknown variables.

The running times were obtained by MATLAB on a SUN ULTRA1 with a 167MHz CPU and 64 MBytes of main memory. To give an idea of the inner loop complexity, we have divided the time of one iteration evaluation by the product of the constraint cardinality and the number of measurements. This time corresponds to a single constraint evaluation, reported on the first line of table 3. The total time for each method in table 3 is proportional to the number of measurements N multiplied by the number of constraints. The results justify our claims regarding efficiency.

Table 3: Timings per constraint and total time for $N = 39$.

Case	1.1	2.2	2.6	3.3	direct
Single constraint [sec]	0.03	0.03	0.03	0.34	0.2
Total [min]	6.4	12.5	38.5	248.0	48.4

5 Conclusion

Eliminating certain measurements during parallel robot calibration seems to be an efficient technique to enhance the robustness of this process, crucial in control and kinematics. Our methods may be applicable to further problems in robot kinematics, including calibration of serial robots, as well as camera calibration. We compare different strategies and conclude that those which work best are not the hardest to compute, from an algebraic point of view. Namely, necessary conditions obtained by direct dialytic elimination improve the initial estimates more accurately than the resultant. Our simulations show a net enhancement over existing direct calibration methods. Our algorithm reduces initial error in pose *determination* by 99% and $80 - 98\%$ respectively, in figures 5 and 3. Furthermore, our algorithm is more *reliable* since it removes the need to use random measured configurations. In fact, boundary configurations behave better than random ones, whereas direct methods fail at the boundary.

Elimination of the *position* parameters is algebraically easier, so it should be possible to carry out a larger part of the computation formally. Our methods should also apply to auto-calibration, an important issue in industrial applications. One approach that we have not fully

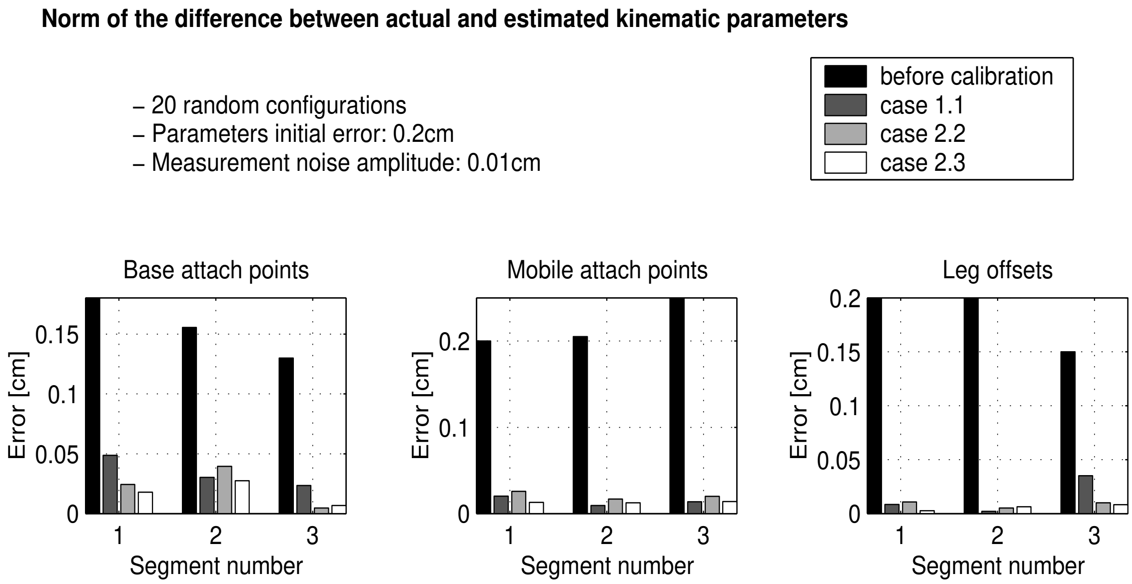

Figure 2: Identification of planar robot geometric parameters

exploited relies on interval arithmetic in order to derive bounds on the determinant value, e.g. with library ALIAS [24]. Further work may focus on calibrating cameras. It is noteworthy that elimination techniques, often known as (re)linearization, has been considered for diverse applications, e.g. cryptography [8].

Acknowledgements: We thank Jean-Pierre Merlet for several insightful discussions, including his trying out our example with ALIAS [24], and Bernard Mourrain for pointing out [19].

References

[1] Baron, L. *Contributions to the estimation of rigid-body motion under sensor redundancy.* PhD thesis, CIM, McGill Univ., 1997.

[2] Bennett, D.J., and Hollerbach, J.M. Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints. *IEEE Trans. Robotics Autom.*, 7:597–606, 1991.

[3] Besnard, S., and Khalil, W. Calibration of parallel robots using two inclinometers. In *IEEE ICRA*, pp. 1758–1763, 1999.

[4] Besnard, S., and Khalil, W. Identifiable parameters for parallel robots kinematic calibration. In *IEEE ICRA*, pp. 2859–2866, Seoul, S. Korea, 2001.

[5] Bondyfalat, D., and Mourrain, B. and Pan, V.Y. Computation of a specified root of a polynomial system of equations using eigenvectors. *Lin. Alg. & Appl.*, 319:193–209, 2000.

[6] Canny, J.F., and Emiris, I.Z. A subdivision-based algorithm for the sparse resultant. *J. ACM*, 47(3):417–451, 2000.

[7] Coleman, T., Branch, M.A., and Grace, A. *Optimization toolbox for use with* MATLAB*: User's guide.* The Math-Works Inc., 1999.

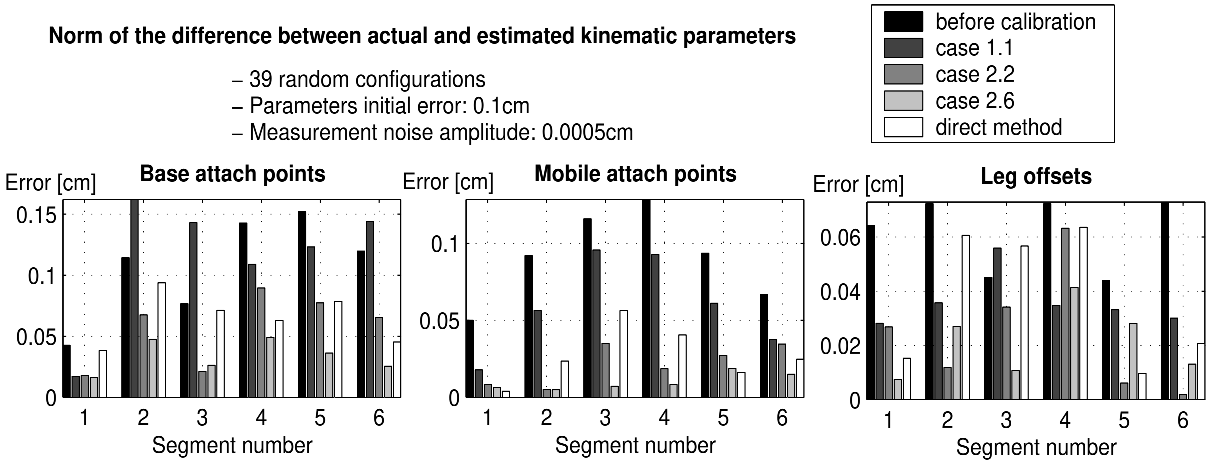

Figure 3: comparison of 4 methods

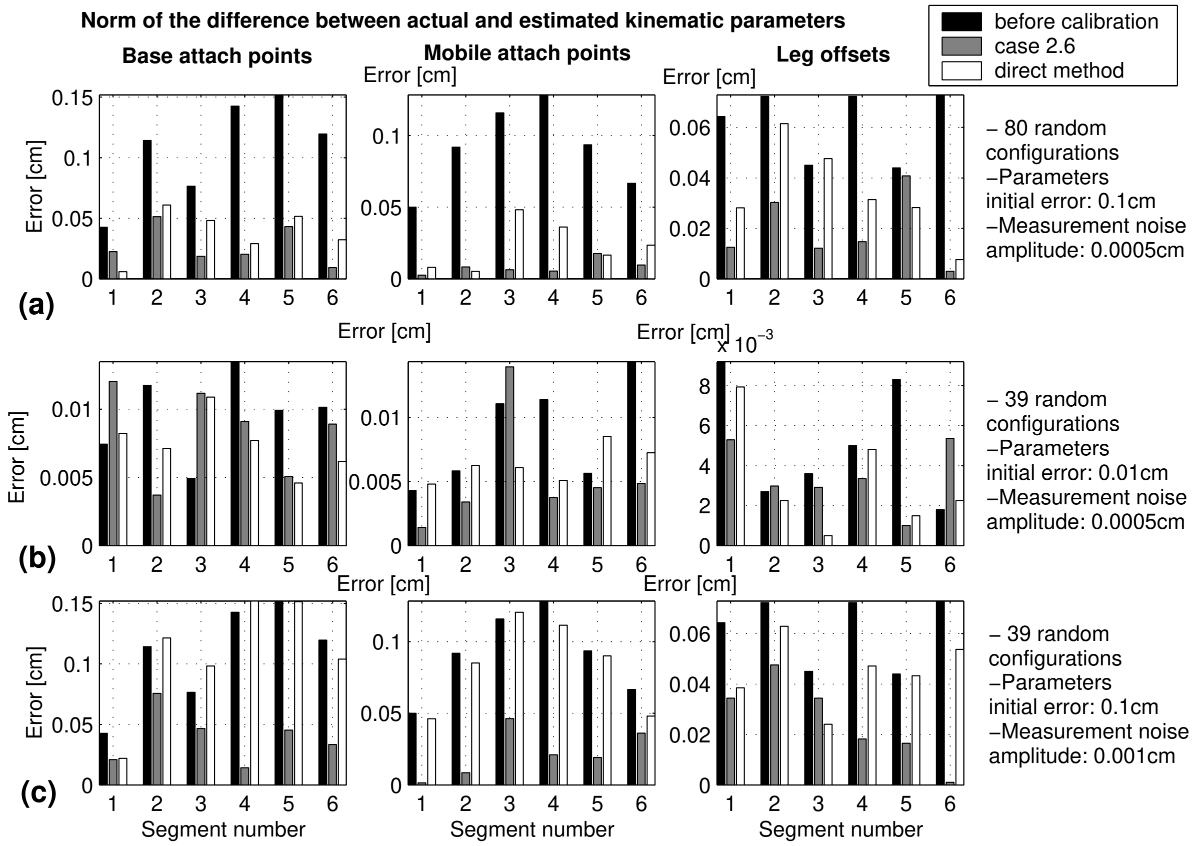

Figure 4: Variations of simulation parameters

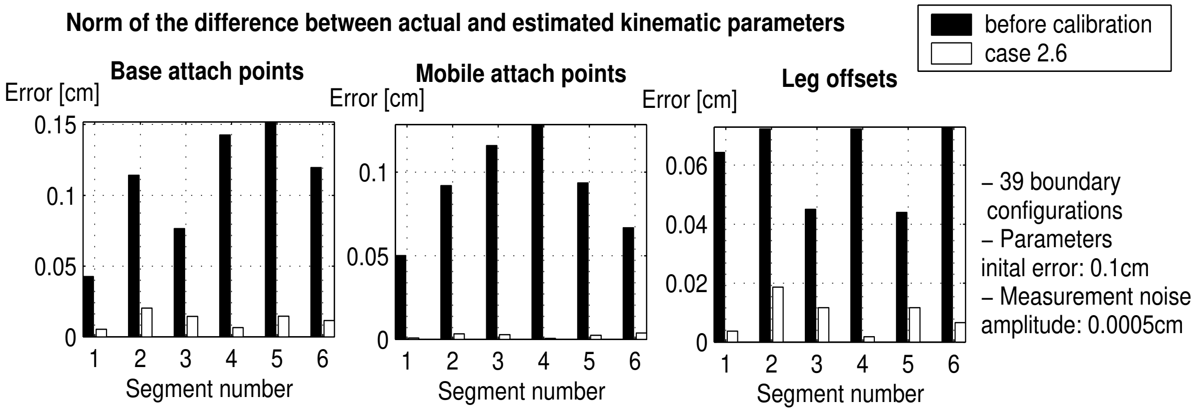

Figure 5: boundary configurations improve calibration in case 2.6

[8] Courtois, N., Klimov, A., Patarin, J., and Shamir, A. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Proc. Eurocrypt*, LNCS 1807, pp. 392–407, 2000.

[9] Corless, R.M., Gianni, P.M., and Trager, B.M. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proc. ACM Symp. Symbolic & Algebraic Comput.,* Maui, Hawaii, pp. 133–140, 1997.

[10] Daney, D. Self calibration of Gough platforms using leg mobility constraints. In *10th World Congress Theory Machines & Mechanisms*, pp. 104–109, Oulu, Finland, 1999.

[11] Daney, D. *Étalonnage géométrique des robots parallèlles.* PhD thesis, École Supérieure des Sciences Informatiques, Univ. of Nice & Sophia-Antipolis, 2000.

[12] Daney, D., and Emiris, I.Z. Robust parallel robot calibration with partial information. In *Proc. IEEE Intern. Conf. Robotics Automation*, pp. 3262–3267, Seoul, S. Korea, 2001.

[13] Daney, D., and Emiris, I.Z. Variable elimination for reliable parallel robot calibration. In *Proc. 2nd Intern. Workshop on Computational Kinematics*, pp. 133–144, Seoul, S. Korea, 2001.

[14] Emiris, I.Z., and Canny, J.F. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symb. Comp.*, 20(2): 117–149, 1995.

[15] Geng, Z., and Haynes, L.S. An effective kinematics calibration method for Stewart platforms. In *ISRAM*, pp. 87–92, 1994.

[16] Husty, M.L. An algorithm for solving the direct kinematics of Stewart-Gough-type platforms. Tech. Report 94-1, McGill CIM, 1994.

[17] Innocenti, C. Algorithms for kinematic calibration of fully-parallel manipulators. In B. Ravani, J-P. Merlet, eds., *Computational Kinematics*, pp. 241–250. Kluwer, 1995.

[18] Khalil, W., and Besnard, S. Self calibration of Stewart-Gough parallel robots without extra sensors. *IEEE Trans. Robotics Autom.*, 15:1116–1121, 1999.

[19] Lazard, D. Résolution des systèmes d'équations algébriques. *Theor. Comp. Sci.*, 15:77–110, 1981.

[20] Lyubeznik, G. Minimal resultant systems. *J. Algebra*, 177:612–616, 1995.

[21] Masory, O., Wang, J., and Zhuang, H. Kinematic modeling and calibration of a Stewart platform. *Advanced Robotics*, 11:519–539, 1997.

[22] Masory, O., H. Zhuang, H., and Jiahua, Y. Kinematic calibration of a Stewart platform using pose measurements obtained by a single theodolite. In *IEEE Intell. Robots & Systems*, pp. 329–334, 1995.

[23] Merlet, J-P. *Parallel robots*. Kluwer, Dordrecht, 2000.

[24] Merlet, J-P. ALIAS: An interval analysis based library for solving and analyzing systems of equations. In *Proc. Conf. Systèmes d'Equations Algébriques*. Toulouse, 2000.

[25] Mourrain, B. The 40 "generic" positions of a parallel robot. In *Proc. ACM Symp. Symbolic & Algebraic Comput.*, Kiev, pp. 173–182, 1993.

[26] Nielsen, J., and Roth, B. Elimination methods for spatial synthesis. In J-P. Merlet, B. Ravani, eds., *Computational Kinematics*, pp. 51–62. Kluwer, 1995.

[27] Oliviers, M.P., and Mayer, J.R.R. Global kinematic calibration of a Stewart platform. *ASME DSC*, 57:129–136, 1995.

[28] Vischer, P. *Improve the accuracy of parallel robots*. PhD thesis, Institut des Systèmes Robotiques, École Polytechnique Fédérale de Lausanne, 1996.

[29] Wampler, C., and Arai, T. Calibration of robots having kinematic closed-loops using nonlinear least squares estimator. In *IFToMM*, pp. 153-158, 1992.

[30] Zhuang, H. Self-calibration of parallel mechanisms with a case study on Stewart platforms. *IEEE Trans. Robotics Autom.*, 13:387–397, 1997.

[31] Zhuang, H., Jiahua, Y., and Masory, O. Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. *J. Robotic Systems*, 15:395–405, 1998.

[32] Zhuang, H., and Roth, Z.S. Methods for kinematic calibration of Stewart platforms. *J. Robotic Systems*, 10:391–405, 1993.