David Daney
Yves Papegay

COPRIN Team, Sophia Antipolis Research Unit
French National Institute for Computer Science
and Control
BP 93, 06902 Sophia Antipolis Cedex, France
yves.papegay@sophia.inria.fr

Blaise Madeline

IMERIR Engineering School of Robotic
and Computer Science
BP 2013, 66011 Perpignan Cedex, France

Choosing Measurement Poses for Robot Calibration with the Local Convergence Method and Tabu Search

Abstract

The robustness of robot calibration with respect to sensor noise is sensitive to the manipulator poses used to collect measurement data. In this paper we propose an algorithm based on a constrained optimization method, which allows us to choose a set of measurement configurations. It works by selecting iteratively one pose after another inside the workspace. After a few steps, a set of configurations is obtained, which maximizes an index of observability associated with the identification Jacobian. This algorithm has been shown, in a former work, to be sensitive to local minima. This is why we propose here meta-heuristic methods to decrease this sensibility of our algorithm. Finally, a validation through the simulation of a calibration experience shows that using selected configurations significantly improve the kinematic parameter identification by dividing by 10–15 the noise associated with the results. Also, we present an application to the calibration of a parallel robot with a vision-based measurement device.

KEY WORDS—robust design, experimentation, calibration, parallel robot, local convergence, Tabu search

1. Introduction

The Gough/Stewart platform offers high accuracy for the manipulation of heavy objects, even at large velocities. This explains its use in a wide range of industrial applications, from flight simulation to the more recent high-speed machine tools.

As a result of defects in manufacturing or assembly, it is well known that the geometry of robotic manipulators does not exactly match the design goals. As a direct consequence, the accuracy of the manipulator is reduced, because robot control heavily relies on a precise description of the kinematic model. One way to tackle this problem is to improve the theoretical kinematic model by means of kinematic calibration. This procedure allows an identification of kinematic parameters through redundant information on the state of the robot provided by measurement, viewed as constraints on the kinematic parameters. The noise associated with sensors implies an error on the calibration results which has to be minimized. To improve the robustness of the calibration methods, there are several known solutions: to increase the number of constraint equations, to take into account knowledge of the noise distribution associated with sensors, and/or to act on the numerical quality of measurements. As shown in Driels and Pathre (1990), the best results are obtained through a judicious choice of the robot configurations used for the measurements.

Several works in robotics have been presented to produce an optimal experimentation plan. The first step is to define different observability indices, associated with a set of poses, and to check their effects on robustness calibration results. Driels and Pathre (1990), Borm and Menq (1991), Khalil, Gautier, and Enguehard (1991), and Nahvi and Hollerbach (1996) propose to observe the identification Jacobian $J_{\mathcal{P}}$ (the derivatives of the constraint equation matrix with respect to the kinematic parameters $\mathcal{P}$) through its singular values (see Section 2). Then an algorithm is used to maximize the observability index as a function of some parameters defining a measurement pose. This leads to the selection of an optimal set of configurations. Many solutions have been proposed. Khalil, Gautier, and Enguehard (1991) use a conjugate-type optimization method. Zhuang, Wang, and Roth (1994) try to avoid local minima by using a simulated annealing approach or a genetic algorithm (Zhuang, Wu, and Huang 1996). Born and Menq (1991) study the determinant of $J_{\mathcal{P}}^{\mathrm{T}} J_{\mathcal{P}}$ to

The International Journal of Robotics Research
Vol. 24, No. 6, June 2005, pp. 501-518,
DOI: 10.1177/0278364905053185

determine how one should modify pose parameters to maximize their index. Linttot and Dunlop (1996) use this algorithm for calibration of a Delta robot. More recently, author seem to be more interested in selecting the best poses inside a large set of measurement configurations (Chiu and Perng 2004). Takeda, Shen, and Funabashi (2004) describe an algorithm for this purpose.

After a brief presentation of the search calibration pose problem (Section 2), the state of the art of optimization methods is shown in Section 3. Then, in Section 4, we propose an algorithm called the iterative one-by-one pose search (IOOPS). This is derivated from DETMAX (Mitchell 1974), widely used in the numerical community for the design of experiments. This algorithm is used to maximize the determinant of $J_{\mathcal{P}}^{\mathrm{T}} J_{\mathcal{P}}$. However, we outline a general version of this algorithm and the reader can adapt it to various observability indices. Additionally, we propose two main improvement of this algorithm. The first is to use a randomization of several indices of observability that permits a multicriteria optimization. The second improvement is to use a Tabu search method to prevent a premature convergence of the algorithm to local minima. Then, in Section 5 we describe its implementation for the specific case of the Gough platform calibration by an implicit method. Finally, a set of optimal configurations is proposed (Section 6) and validated by a realistic calibration simulation (Section 7). In Section 8 we present two applications: a real experimentation of calibration, which demonstrates the benefit of choosing the measurement configurations. A discussion (Section 9) concludes this paper.

2. Calibration and Observability Index

The aim of calibration is to improve the position accuracy of a robot. Hence, it is necessary to obtain redundant information on the state of the robot. For this purpose, we may use measurements provided by internal sensors placed along various joints of the robot, external information obtained by measurement machinery, and/or additional constraints on the end-effector or joints. These measurements $\mathcal{M}$ are related to the kinematic parameters $\mathcal{P}$ via constraint equations $F(\mathcal{P}, \mathcal{M})$. The first problem is to obtain a number of constraints N_c greater than or equal to the number of kinematic parameters $N_{\mathcal{P}}$ to be identified. Note that for a given pose ζ of the robot, the dimension of F, which measures the amount of redundant information on the state of the robot, is generally less than the number of unknowns $N_{\mathcal{P}}$. Therefore, to obtain the minimal number of constraints necessary to identify the kinematic parameters, it is necessary to collect information from the robot in a variety of different poses. We thereby obtain a system of constraint equations of the following form:

$$\mathcal{S} = \begin{cases} F_1(\mathcal{P}, \mathcal{M}_1[\zeta_1]) \\ \dots \\ F_N(\mathcal{P}, \mathcal{M}_N[\zeta_N]) \end{cases} \tag{1}$$

where $N_c = N \times \dim(F_j) \geq N_{\mathcal{P}}$, $[\zeta_1, \dots, \zeta_j, \dots, \zeta_N]$ denote the poses in which the robot is observed, and $\mathcal{M}_j[\zeta_j]$ is the data set corresponding to the pose ζ_j. For the actual kinematic parameters $\hat{\mathcal{P}}$ of the robot, the equations $F_{1,\dots,N}$ should be approximately zero. By solving the equations $\mathcal{S}$ we may identify the kinematic parameters $\mathcal{P} = \hat{\mathcal{P}}$. Because of sensor noise and instability introduced by physical constraints, however, the measurements $\mathcal{M}_{1,\dots,N}$ are inexact. Measurement errors are propagated in the resolution machinery and affect the resulting values obtained for the parameters $\mathcal{P}$. One strategy for improving the robustness of the calibration with respect to measurement noise is to obtain an overconstrained system of equations by increasing N. In practice, this technique yields improved robustness only while the number N of configurations remains less than a certain threshold value (see Nahvi, Hollerbach, and Hayward 1994). We may use the identification Jacobian $J_{\mathcal{P}}$ to ensure that our calibration is robust with respect to measurement noise. This matrix is defined as the derivative of the constraining equations F_j with respect to the kinematic parameters $\mathcal{P}$. To the identification Jacobian, in turn, we may associate an index of observability O to characterize the sensitivity of constraint equations to variation in the kinematic parameters. The matrix $J_{\mathcal{P}}$ is a function of the given measurement configuration; by choosing a pose inside the workspace for which the observability index $O(J_{\mathcal{P}})$ is optimized, we may improve the robustness of the calibration method.

Many indices of observability have been proposed. Born and Menq (1991) use the determinant of $J_{\mathcal{P}}^{\mathrm{T}} J_{\mathcal{P}}$, while Driels and Pathre (1990) and Khalil, Gautier, and Enguehard (1991) use the condition number of the identification Jacobian. More generally, Nahvi and Hollerbach (1996) define the following indices and others in terms of singular values (denoted by $\sigma_L \leq \dots \leq \sigma_1$) obtained from a singular value decomposition of $J_{\mathcal{P}}$.

- $O_1 = \sqrt[L]{\sigma_L \dots \sigma_1}/\sqrt{m}$ where m is the number of measurement poses and L is the number of singular values. This index corresponds to the determinant of $J_{\mathcal{P}}^{\mathrm{T}} J_{\mathcal{P}}$; here, $\sqrt{\det(J_{\mathcal{P}}^{\mathrm{T}} J_{\mathcal{P}})} = \sigma_L \dots \sigma_1$. The sensitivity of F with respect to variations of the $\mathcal{P}$ is increased when O_1 is maximal.

- $O_2 = \sigma_L/\sigma_1$, the inverse of condition number of J (see Driels and Pathre 1990). This index has a maximal value equal to 1. Maximizing this index permits us to ensure that no singular value is favored over another or, in physical terms, that the errors incurred in determining the kinematic parameters are "homogenized".

- $O_3 = \sigma_L$ and $O_4 = \sigma_L^2/\sigma_1$ are slight modifications of O_1 and O_2 introduced in Nahvi and Hollerbach (1996).

3. Formulation of the Problem

In this section, we define the optimal calibration pose problem as an optimization problem under constraints. Then, an overview of the possible solving method is given.

3.1. Aim

As shown in Section 2, many observability indices permit us to evaluate the numerical quality of an identification Jacobian. Then, the problem is to determine an algorithm to improve (maximize in our case) these indices.

Two set of parameters define the identification Jacobian matrix $J_{\mathcal{P}} = \partial F(\mathcal{P}, \mathcal{M})/\partial \mathcal{P}$: the kinematic parameters $\mathcal{P}$ and the measurement parameters $\mathcal{M}$.

The kinematic parameters $\mathcal{P}$ are given whatever the state of the robot. In robotic calibration problems, their values are not exactly known. However, it is realistic to hypothesize that the initial estimations of $\mathcal{P}$ given by the constructor are well done. Then, to improve the numerical quality of the identification Jacobian for actual kinematic parameters is very close to solving the problem of the calibration optimal pose search with a good estimation of their values.

Only the chosen measurement configurations of the robot may be changed experimentally to improve the indices associated with the identification Jacobian matrix. Let ζ_i be the minimal vector, which defines one measurement configuration $i = 1, \ldots, N$. It is possible to select this subset ζ_i from the set of measurement parameters $\mathcal{M}_i$ to define the robot pose.

The choice of the vector ζ_i inside $\mathcal{M}_i$ is important, because the other parameters (called η_i) modeling the measurement vector $\mathcal{M}_i = \{\zeta_i, \eta_i\}$ have to be determinated as a function of the kinematic parameters and ζ_i: $\eta_i = h(\zeta_i, \mathcal{P})$. Then, this choice has to be motivated and adapted to the calibration method.

The problem formulation is now to determine $\zeta = \{\zeta_1, \ldots, \zeta_N\}$ such that the objective function $O(J_{\mathcal{P}}(\zeta, \eta, \mathcal{P}))$ with $\eta_i = h(\zeta_i, \mathcal{P})$ is maximal. Then, the experiment design consists of putting the robot in (or near) $i = 1, \ldots, N$ configurations ζ_i and collecting the measurement information $(\mathcal{M}_i)$.

The difficulty of this problem is that the number of parameters ζ is large compared to only one objective function $O(J_{\mathcal{P}})$. However, we can decrease the search area of ζ by adding some constraints. Hence, many natural physical equations or inequalities (called $g_j(\mathcal{P}, \zeta)$, $j = 1, \ldots, M$) may be introduced. In robotics calibration, these constraints may be the workspace, the assembly mode of the robot or the mechanical constraints used by a self-calibration method (for example, the fixation of the end-effector or the segments mobility).

The formulation is given as

$$
\begin{array}{rl}
\mathbf{Max}_{\zeta} & \mathcal{C}(\zeta, \mathcal{P}) \\
\textbf{with} & \mathcal{C}(\zeta, \mathcal{P}) = O(J_{\mathcal{P}}(\zeta, \eta, \mathcal{P})) \\
& \left\{ \begin{array}{l} \zeta = \{\zeta_1, \ldots, \zeta_N\}, \\ \eta_i = h(\zeta_i, \mathcal{P}), \end{array} \right. \text{for } 1 \le i \le N \\
\textbf{subject to} & \\
& g_j(\mathcal{P}, \zeta) = 0 \text{ for } 1 \le j \le m \\
& g_j(\mathcal{P}, \zeta) \le 0 \text{ for } m+1 \le j \le M.
\end{array} \tag{2}
$$

3.2. Deterministic Local Methods

The classical algorithm to solve the problem given by eq. (2) is to use an optimization method as a sequential quadratic programming (SQP) method (see Lemarchal et al. 2003).

If the objective function $\mathcal{C}$ and the constraints are linear, it is possible to use many well-known algorithms, such as the Simplex or the Interior-point algorithms. If the objective function is quadratic and the constraints are linear, the quadratic problem (QP) consists of transforming that problem into the solving of an equivalent system of equations, called the Karush–Kuhn–Tucker (KKT) equations, using Lagrange multipliers. This system is then solved by a Newton-like approach. The SQP algorithm used to solve the problem (2), when the objective function and the constraints are non-linear, consists of linearizing the equations and processing iteratively the simplified subproblems by a QP algorithm. In our case, the Jacobian and the Hessian are given by a numerical approximation.

These methods have the major disadvantage that they are sensible to the local minimum. Additionally, the huge number of parameters (ζ) that we want to identify does not permit a good convergence of this type of algorithm. In the next subsection, many solutions are proposed to tackle this problem.

3.3. Reformulation of the Problem: the Iterative One-by-one Pose Search

When using an optimal configuration search, it is difficult in general to find all the parameters defining the N configurations of measurement in a single step. This is due to the number of parameters, which is equal to $N \times \mathrm{Dim}(\zeta_i)$ with N being the number of poses and $\mathrm{Dim}(\zeta_i)$ the number of components defining one measurement configuration.

One solution is to subdivide the problem. The principle is to search one-by-one each pose ζ_i. Then this process is executed iteratively to improve the interaction between each pose and then to maximize the numerical quality of the identification Jacobian matrix.

We propose to adapt the standard DETMAX algorithm (Mitchell 1974) for experimental design (Walter and Pronzato 1997). The frame of this algorithm consists of selecting one additional line (or block-line) of the identification matrix $J_{\mathcal{P}}(\zeta, \eta, \mathcal{P})$ corresponding to one additional pose ζ_{N+1}.

Then, this will improve the determinant of the matrix by searching ζ_{N+1} by an optimization method under constraints (or inside a finite set). Furthermore, it discards the configuration ($\zeta_i, i = 1, \ldots, N+1$) such that the resulting determinant is maximal. The process is iterated until the same configuration is consecutively added and removed.

The DETMAX algorithm is dedicated to improve the determinant of a matrix. The operations of addition and removal of a block-line yield reduced complexity and runtime.[1] However, it is also well adapted for any observability index defined in Section 2. To avoid confusion, we prefer to rename the frame of the DETMAX algorithm the IOOPS. Details of the algorithm are given in Section 4.1.

This algorithm simplifies the search calibration pose problem but does not avoid the following two problems:

- the local convergence problem due to the use of an optimization method under constraints;
- the difficulty of processing with a multicriterion optimization (see Sections 4.1.8 and 4.2).

The meta-heuristic methods may help us to circumvent these problems.

3.4. Meta-heuristic Methods

In combinatorial optimization, we can find different types of methods. The first types are called deterministic methods. These use tree search methods and eventually branch and bound. The second types are usually used when the search space is too wide. These are incomplete and gather local search (simulated annealing, Tabu search, MinConflict, etc.) and population-based search (genetic algorithms, Go With The Winner, etc.). All these incomplete methods are also called meta-heuristics.

Meta-heuristics are not systematic; they cannot guarantee finding a solution if it exists, nor show that there is no solution, if necessary. They generally use stochastic processes, i.e. they use random processes during the search in order to introduce diversification into the exploration of the search space. These methods appear generally effective, and make it possible to deal with problems too large to be treated effectively by constructive methods (i.e. tree search methods). The interested reader should refer to Hao, Galinier, and Habib (1999). The local search methods do not proceed by construction of a solution but by successive improvements of an initial solution. Several tools are necessary. First, a neighborhood function will make it possible to move in the search space starting from the initial solution. Then an evaluation function will allow us to choose among the neighbors. One can model the general procedure of most local methods, as follows.

1. Choose an initial solution s.
2. Choose among the neighbors of s a solution $\bar{s}$.
3. Check if the termination condition is true.
4. Decide if $\bar{s}$ must replace s; return to 2.

According to the method, the last three points will follow different strategies. For the presented local search methods, we call s the current solution, $\bar{s}$ the new solution taken in the neighborhood $N(s)$, and finally $E(s)$ the function that computes the cost of the solution s.

3.4.1. Genetic Algorithm

The genetic algorithms introduced by Holland (1975) belong to the class of the algorithms known as stochastic algorithms. Indeed, most of their operations are based on chance. However, this chance is directed by the evaluation function, which makes it possible to introduce into the genetic operators a useful quantity of determinism in order to obtain a solution. The total diagram of a genetic algorithm consists of six principal elements:

1. a set of initial configurations called the population;
2. a function for chromosomes coding/decoding;
3. genetic operators (mutation, crossing-over, etc.);
4. an evaluation function;
5. a selection algorithm (Tournament, Roulette, etc.);
6. parameters.

The formal algorithm is described as follows:

```
t=0 Initialization of the population P(t)
Evaluation of the individuals of P(t)
while (Termination Condition) do
    t ← t+1
    Selection, copy of parents of P(t-1) into P(t)
    Application of the genetic operators on P(t)
    Evaluation of the individuals of P(t)
end
Decode the best solution and return it.
```

Zhuang, Wu, and Huang (1996) have already used a genetic algorithm to optimize robot calibration poses.

The interested reader can usefully consult Baeck, Fogel, and Michalewicz (1997).

3.4.2. Simulated Annealing

Simulated annealing is a local method inspired by thermal annealing used in the metallurgical industry to obtain a perfect

[1] The reader should consult Walter and Pronzato (1997) for more information.

crystal. The industrial technique used consists of gradually lowering the temperature of metal in fusion, so that by cooling the molecules slowly a stable configuration is reached. The algorithm that takes as a starting point this method was introduced by Kirkpatrick, Gelatt, and Vecchi (1983). The principle of simulated annealing is as follows. A solution by chance is chosen s, and the temperature is fixed. Then we enter the loop of the algorithm.

- Choose randomly a neighbor $\bar{s}$.
- Calculate the difference between the evaluation of s and $\bar{s}$: $\delta = E(s) - E(\bar{s})$.[2]
- $s \leftarrow \bar{s}$ with a probability $e^{\Delta/T}$ if δ is negative, 1 if not.
- Decrease the temperature.

Different criteria from stop can be considered: when the reductions in temperature prove to be ineffective, when a certain number of movements are carried out, etc. Generally, one starts by using a very high temperature, which it will be necessary to decrease by successive steps during the execution. The difficulty is to find a decreasing temperature function that is effective. This algorithm keeps the new solution systematically if it improves the current solution; if not, the new solution, which degrades the current solution, is selected with a probability that decreases over the course of time. Therefore, at the beginning of the execution, one tends to choose any neighbor, then focusing itself gradually on a subspace of the search space by accepting the solution that degrades less and less the current solution. The advantage of this method compared to the method of descent is that it can leave the local optimum (as long as the temperature is sufficiently high). However, the choice of the initial temperature and the function of decrease associated with this temperature remain parameters that are rather difficult to regulate.

Zhuang, Wang, and Roth (1994) also contribute by using this method.

3.4.3. Tabu

The Tabu search, introduced by Glover (1986), is a method which seeks to guide the search in an intelligent way through the solution space. With this intention, it has a memory (the Tabu list) of the already found solutions to avoid loopings. The Tabu search can be summarized as follows. We start by initializing s, using for example a greedy algorithm, and initializing the memory with vacuum. Then, entering the loop of the algorithm

- search $\bar{s} \in V(s)$ such that $\bar{s}$ minimizes $E(\bar{s})$[3] and then $\bar{s} \notin M$;
- update the memory M by adding to it s and deleting the oldest element.

The stop criterion can be the authorized maximum number of moves, to have reached an acceptable cost, not to have obtained improvement since a certain number of movements; Tabu research will thus seek the best neighbor not tabu, even if this degrades the current solution. A solution that was selected will be memorized, and prohibited (tabus) during a certain general number of iterations. This will avoid loopings and moreover allow the algorithm to leave the local optimum. Many parameters will influence the effectiveness of the Tabu method. The determinant parameters are the size of the Tabu list, the neighborhood function, the portion of the visiting neighborhood if it is too large, and the initialization function. There are many variations of Tabu research. One can, for example, not memorize the last solutions but the couples value/variable which changed, one prohibits thus unit of solutions (all those which contain this instantiation). One can also integrate the random one in the choice of the neighbors, or set up criteria of aspiration, when the criteria of prohibition of the list are too demanding. The reader can consult Glover and Laguna (1997) for a complete description of Tabu search.

We fully explain the application of Tabu search in our algorithm in Section 4.2.

4. Proposed Solutions

In the following subsection we describe the principle of the IOOPS algorithm. In the next subsections we give precise details of the algorithm in the basic case. Then we propose some improvements to take into account the multicriterion and local convergence problems.

4.1. Iterative One-by-one Pose Search

4.1.1. Measurement Search Area

We have to define the search area where the optimal calibration poses may be found. We consider two cases of search area type, as follows.

- Let Ω_d be a finite set $[\zeta_1, \ldots, \zeta_N, \ldots, \zeta_W]$ of robot configurations (with $W > N$).
- Let Ω be an infinite set of robot poses but bounded. For this case, Ω is defined by several equalities and inequalities. For example, a robot pose ζ is an element of Ω if the constraints (equalities or inequalities) $g_{1,\ldots,M}(\zeta)$ are true or equal to zero. In robotics, this area may be defined by some constraints on articulate/generalized coordinates, measurement machinery workspace, and/or mechanical calibration constraints due to fixture system.

[2] It is appropriate that we seek to minimize $E(s)$.

[3] Note that $E(\bar{s})$ can be worse than $E(s)$.

4.1.2. Notations

We use the following conventions.

- N is the number of measurement poses used to calibrate the robot.
- ζ_i denotes the vector of parameters defining the ith configuration of measurement.
- $\Upsilon^N = [\zeta_1, \dots, \zeta_N]$ is the set of parameter configurations associated with N configurations of measurement.
- Ω is the workspace of all possible configurations of measurement (two types of search area are defines in Section 4.1.2).
- $\mathcal{C}_a$ and $\mathcal{C}_r$ are indices associated with the N (or $N+1$) measurement configurations Υ^N (or Υ^{N+1}). These indices will be the observability $\mathcal{C}_a, \mathcal{C}_r = O_1, O_2, O_3$ or O_4. Precisely $\mathcal{C}_a(\Upsilon) = O(J_{\mathcal{P}}(\Upsilon, \eta, \mathcal{P})$ as the definition given in Section 2. These are the indices that we want to maximize.
- ζ^+ = AddFind(Ω,Υ^N,$\mathcal{C}_a$) returns the configuration ζ^+ chosen inside Ω, which maximize the criterion $\mathcal{C}_a$ defined as a function of Υ^N.
- ζ^- = RemoveFind(Υ^{N+1},$\mathcal{C}_r$) returns the configuration ζ^-, one component of Υ^{N+1}, which maximize the criterion $\mathcal{C}_r$ defined as a function of Υ^{N+1}.

4.1.3. Iterative One-by-one Pose Search Algorithm Frame

The frame of the IOOPS algorithm is shown in Figure 1. For a given number N of measurement poses, this process is a loop where each iteration is defined by two steps.

Step 1 `Add a new configuration` $N+1$. Find a virtual new configuration $N+1$ denoted ζ^+ inside Ω, which maximizes the criterion $\mathcal{C}_a$. In the case of calibration pose search, this consists of adding a new “block" line to the Jacobian matrix $J_{\mathcal{P}}$ and computing the pose parameters defining this(these) line(s) to maximize one of the observability indices $O_{1,2,3,4}$.

Step 2 `Remove one of the` N+1 `configurations`. Remove one of the $N+1$ poses denoted ζ^- such that the criterion $\mathcal{C}_r$ calculating the set $\{\Upsilon^{N+1} - \{\zeta^-\}\}$ is maximized. This consists of removing the new “block" line to the Jacobian matrix $J_{\mathcal{P}}$ such that one of the observability indices $O_{1,2,3,4}$, computed for the Jacobian matrix after removal, is maximized.

4.1.4. Termination Condition

For the basic algorithm, the termination condition is true if $\zeta^+ = \zeta^-$; i.e. the same “block" lines of the Jacobian are “added” and “removed” in one iteration. This condition of termination is not demonstrated but, experimentally, few iterations (around N) are necessary (see Walter and Pronzato 1997). We never observe the non-termination of this algorithm and the execution time never exceeds a few minutes.

```
define $N$, $\Omega$.
init $\Upsilon^N \in \Omega$, $\mathcal{C}_a$,$\mathcal{C}_r$.
while (Terminaison Condition) do
    $\zeta^+$=AddFind($\Omega$,$\Upsilon^N$,$\mathcal{C}_a$)
    $\Upsilon^{N+1} = \Upsilon^N \cup \zeta^+$

    $\zeta^-$=RemoveFind($\Upsilon^{N+1}$,$\mathcal{C}_r$)
    $\Upsilon^N = \Upsilon^{N+1} - \{\zeta^-\}$

end
```

Fig. 1. IOOPS Algorithm.

```
for $i = 1, \dots, U$ do
    to choose $\zeta_i$ in $\Omega$
end
To select $\zeta^+$ such that
    $\max_{\zeta^+=\zeta_i}([\mathcal{C}_a(\Upsilon^N + \{\zeta_1\}), \dots, \mathcal{C}_a(\Upsilon^N + \{\zeta_U\})\}])$
```

(a) Ω is finite

```
for $i = 1, \dots, U$ do
    to initialize $\zeta_i$ in $\Omega$
    $\underset{\zeta_i}{\textbf{Max}}\ \mathcal{C}_a(\zeta_1, \dots, \zeta_N, \{\zeta_i\})$
      Subject to $\zeta_i \in \Omega$
        $g_j(\zeta_i) = 0$ for $j = 1, \dots, m$
        $g_j(\zeta_i) \leq 0$ for $j = m+1, \dots, M$
end
To select $\zeta^+$ such that
    $\max_{\zeta^+=\zeta_i}([\mathcal{C}_a(\Upsilon^N + \{\zeta_1\}), \dots, \mathcal{C}_a(\Upsilon^N + \{\zeta_U\})\}])$
```

(b) Ω is infinite but bounded

Fig. 2. The AddFind algorithm.

4.1.5. AddFind Algorithm

The AddFind algorithm is shown in Figure 2. This procedure is specified as a function of the definition of Ω.

To select ζ^- such that

$$\max_{\zeta^-=\zeta}([\mathcal{C}_r(\Upsilon^{N+1}-\{\zeta_1\}),\ldots,\mathcal{C}_r(\Upsilon^{N+1}-\{\zeta_{N+1}\})\}])$$

Fig. 3. The RemoveFind algorithm.

If Ω is a finite set of poses (see Figure 2(a)), we first select U poses inside Ω. We can select all poses of Ω ($U = \dim(\Omega)$) or a subset of U poses randomly chosen in Ω. Then AddFind consists of checking what pose $\zeta^+ = \zeta_i$, $i = 1, \ldots, U$ maximizes the index $\dashv$ calculated for the N initial poses $\Upsilon^N = [\zeta_1, \ldots, \zeta_N]$ additionally to ζ^+.

If Ω is a bounded search area (see Figure 2(b)), the algorithm described in Section 3.2 is used (denoted **Max**) to find inside Ω a configuration ζ^+ which maximizes $\mathcal{C}_a(\Upsilon^N + \{\zeta^+\})$. The constraints $g_{1,\ldots,M}$ permit us to focus the search algorithm inside a realistic measurement area. The local convergence of the SQP method implies that the initial estimation of ζ^+ influences the result. To avoid this problem, we process U executions of the SQP algorithm with U initial estimations; the best result provided by **Max** is then kept. We have three versions to initialize ζ_i (see Figure 2(b)). We can choose the initial estimation of ζ_i randomly inside Ω, inside a subset defined by uniformly distributed poses in Ω, inside a list of interesting configurations (see Section 6).

4.1.6. RemoveFind Algorithm

The RemoveFind algorithm is shown in Figure 3. For the basic version of IOOPS, the criterion $\mathcal{C}_r$ is, in fact, equal to $\mathcal{C}_a$. We will see in the following section that this is not always the case.

4.2. Randomization of the Iterative One-by-one Pose Search

The first idea is to introduce a randomization into the IOOPS algorithm to obtain a better solution by scanning a larger portion of the search space. We can introduce randomization at several points:

- in the initialization process;
- in the criterion choice;
- in the deleting process.

We see in Section 6.2 the better tuning that we have found. This randomization allows us to find the best way of IOOPS for the initialization, the criterion choice, and the deleting process.

To select ζ^- such that

$$\max_{\zeta^-=\zeta}([\mathcal{C}_r(\Upsilon^{N+1}-\{\zeta_1\}),\ldots,\mathcal{C}_r(\Upsilon^{N+1}-\{\zeta_{N+1}\})]-\{\zeta_i^+,\ldots,\zeta_j^+\})$$

Fig. 4. The RemoveFind algorithm with the Tabu algorithm.

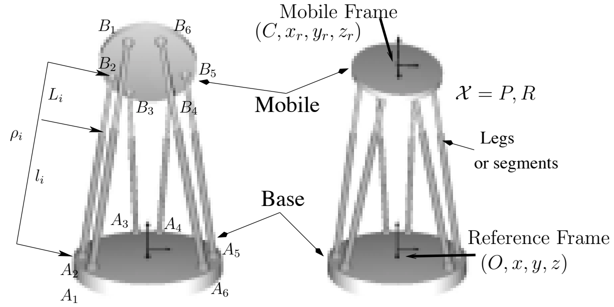


Fig. 5. The Gough platform.

4.3. Improvement by Applying a Tabu Search

The idea is to benefit from the discretization of the search space done by the IOOPS algorithm. Introducing a Tabu search process into the IOOPS can be easily done. We memorize the n last ζ^+ in a Tabu list $T = \{\zeta_i^+, \ldots, \zeta_j^+\}$ of size n such that $j - i = n$. The RemoveFind algorithm is now as in Figure 4.

Obviously, the termination condition is no longer valid. The new termination condition is now the number of iterations allowed. The classical IOOPS stops in a few iterations, but the Tabu search process allows us to reach a better solution by locally degrading the current solution and following a better convergence. Thus, it is necessary to allow a larger number of iterations to obtain good results.

5. An Example

5.1. Modeling and Kinematics of the Gough Platform

In this section we describe the Gough platform and its kinematic models. The manipulator consists of two rigid bodies: the base and the mobile platform, connected by six legs or segments. The leg linear actuators provide six degrees of freedom for the platform pose relative to the base, corresponding to position P and rotation matrix R. A configuration $\mathcal{X} = [P, R]$ is associated with six length variations L_i measured by "internal" leg sensors, $i = 1, \ldots, 6$.

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 (Vischer, 1996). However, as shown in Masory, Wang, and Zhuang (1997), the principal source of error in positioning is due to limited knowledge of the joint centers and to the fact that part of the length is not given by the sensors. We thus use a simpler model with attachment points a_i in the reference frame, b_i in the mobile frame, and offset lengths l_i. This gives seven parameters per leg, and therefore 42 overall, denoted by $\mathcal{P}$.

Kinematic models are essential for controlling the robots, and yield the constraints used in calibration. Inverse kinematics express the length variation for the ith leg as a function of $\mathcal{X}$:

$$\|P + Rb_i - a_i\|_2 = L_i + l_i,\ i = 1 \dots 6. \tag{3}$$

To describe the rotation matrix, we use the Euler angle parameters q_1, q_2, q_3. For the Gough platform, the forward kinematics model is more difficult to compute, since it consists of solving (3) for P and R given L_i and $\mathcal{P}$. It has been shown how techniques from computer algebra based on Groebner bases can be used in principle to solve this problem. From a practical point of view, this is however much more tricky. Faugère and co-workers kindly gave us access to the implementation of their recent work (Rouillier et al., 2005), which addresses the practical issues and allows one to efficiently solve this problem.

5.2. Implicit Calibration Method

We test the method of optimal configuration search on the parameter Jacobian provided by a basic calibration method. The implicit method (Zhuang, Jiahua, and Masory 1998) (or inverse method) presents the more popular method to calibrate a parallel robot. It consists of estimating the pose error indirectly through the leg length error. The constraint equations are directly provided by the implicit inverse kinematic equations (4)

$$F_i^{IK} = \|P + Rb_i - a_i\|^2{}_2 - (L_i + l_i)^2,\ i = 1, \dots, 6, \tag{4}$$

where the unknowns are 7 × 6 kinematic parameters $\mathcal{P}_i = [a_i, b_i, l_i]$ with $i = 1 \dots 6$ the index of the segment, and measurements are $\mathcal{M}_j = [P_j, R_j, L_{i,j}]$ with $j = 1, \dots, N$ the index of the configuration. Each constraint follows from eq. (4) and depends only on the seven parameters of the respective leg. So, we can decouple the robot calibration in six subproblems and take $N \geq 7$ measures. In this paper, we use a least-squares method (using function Leastsq of MATLAB, implementing a Levenberg–Marquart algorithm) to solve these equations in terms of the parameters. Namely, we solve the following minimization problem, for each leg $i = 1..6$:

$$\text{Minimum} \textstyle\sum_{j=1}^{N} \mathcal{C}_{i,j}$$
$$\mathcal{C}_{i,j} = [F_{i,j}^{IK}(\mathcal{P}_i, \mathcal{X}_j, L_{i,j})]^{\mathrm{T}}[F_{i,j}^{IK}(\mathcal{P}_j, \mathcal{X}_j, L_{i,j})].$$

To find out in which direction we should look for the minimum, we compute the Jacobian matrices corresponding to the subcalibration of each leg i independently: $J_{\mathcal{P}_i} = \partial F_{i,j=1..N}^{IK} / \partial \mathcal{P}_i$. Each line $j = 1, \dots, N$ of this matrix (see Zhuang, Jiahua, and Masory 1998) is done by

$$\left\{ -P_j + R_j b_i - a_i{}^{\mathrm{T}}, \left[R_j^{\mathrm{T}}(P_j + R_j b_i - a_i) \right]^{\mathrm{T}}, -(L_{i,j} + l_i) \right\}. \tag{5}$$

The aim of the measurement pose search problem is to find the configurations which maximize an observability index of the matrix $J_{\mathcal{P}_i}$ for the calibration of one leg i of the Gough platform. This process is repeated for all legs $i = 1 \dots 6$. In this case, we fix the values of the kinematic parameters to a given value, such as that provided by the robot constructor. However, to observe the influence of the same set of poses to calibrate all robot legs in one step, we introduce the Jacobian $J_{\mathcal{P}}$ where the diagonal consists of the parameter Jacobian $J_{\mathcal{P}_i}$ of each leg. Each block matrix $J_{\mathcal{P}_i}$ is built for each vector of kinematic parameters $\mathcal{P}_i$ but with the same measurement data, as

$$J_{\mathcal{P}} = \begin{pmatrix} J_{\mathcal{P}_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & J_{\mathcal{P}_6} \end{pmatrix}_{6N_c \times 42} . \tag{6}$$

The algorithm IOOPS, applied on the matrix $J_{\mathcal{P}}$, permits us to improve an associated observability index, and then to improve the identification of kinematic parameters by minimizing the measurement noise influences. We must warn that, in this paper, we prefer to search the set of measurement poses that improve the calibration of all legs and not six independent sets of measurement poses that improve the numerical qualities of each legs calibration (in this case, the algorithm has been applied on each matrix $J_{\mathcal{P}_i}$; see Daney 2002). This choice permits us to divide by six the number of poses used by the experimentation.

5.3. Adaption and Implementation of the Iterative One-by-one Pose Search Algorithm

In this problem, we consider the hypothesis that the actual robot kinematic parameters are close to an initial estimation given by the robot constructor. Then, the optimal configurations that improve the numerical qualities of a calibration process, and are calculated with the constructor estimation of the kinematic parameters, are close to the optimal configuration given for the actual parameters.

We formally define the vector of configuration parameters ζ introduced previously. We could have chosen to use articular coordinates, but instead we prefer to encode configuration parameters using the position and the orientation (Euler angles) of the robot $\zeta = \mathcal{X} = [P_x, P_y, P_z, q_1, q_2, q_3]$ (see Section 5.1). The reason is that generalized coordinates are well suited for defining the identification Jacobian associated with the calibration method presented in Section 5.2 to test this algorithm. In the other case, where one pose is defined by its articular coordinates, a forward kinematic is needed to compute the elements of the Jacobian $J_{\mathcal{P}}$. This is costly in execution time and prone to error if we use a Newton-like algorithm. Then, the parameters that define one pose of the robot are the generalized coordinates. The other parameters, the leg length $L_{i,j}$, are deduced by using the inverse kinematics.

The measurement search area is defined by the articular workspace $L_{min} \leq L \leq L_{max}$, the sign of the inverse kinematic Jacobian matrix (the assembly mode), and the direction of vector defining the mobile frame (to obtain several realistic measurement poses). Additionally, it may be interesting to add some constraints applying to generalized coordinates as a restrictive definition of the robot workspace. These constraints are modeled by the inequalities g_j, $j = 1, \dots, M$, which are used by the SQP algorithm **Max** in the procedure AddFind (see Figure 2(b)). In another type of calibration method, such as a self-calibration with mechanical constraints, these additional constraints are the means to take into account the mechanical constraints. Another use of these additional constraints is to avoid looking for configurations in the neighborhood of singularities of the robot or of the legs of the robots. Typically, by excluding parts of the workspace from the search area through such additional constraints, we are able to handle the case of a robot with leg singularities, such as Delta robots (Lintott and Dunlop 1996).

Our algorithm is implemented in MATLAB. The Jacobian $J_{\mathcal{P}}$ is initialized using such N randomly chosen configurations, such N uniformly distributed configurations inside the measurement workspace (MW) or N configurations chosen inside a special list; they are denoted by $\Upsilon_N = [\zeta_1, \dots, \zeta_N]$. The type of chosen poses and their number N is discussed in Section 6.

As the same process, the procedure AddFind is executed U times. For each execution, a pose, denoted ζ^U, initializes the SQP algorithm (see Section 3.2). This pose is chosen randomly inside the workspace, or in a list of poses chosen uniformly in the workspace, or in a specific list described below. Then the U poses determinated by the SQP algorithm, such that the criterion is maximized, are stocked in a list. As a result of the local convergence of this algorithm, the U proposed poses are different as a function of the initialization of SQP. In fact, only the selected pose that maximizes at most the criterion is kept; the pose is denoted ζ^+ and put in place of ζ^{N+1}.

Finally, the procedure RemoveFind consists of constructing $i = 1, \dots, N$ Jacobian matrices $\{J_{\mathcal{P}}^1, \dots, J_{\mathcal{P}}^N$ as a function of $\zeta_1, \dots, \zeta_N + 1$ less ζ_i. Then we check the N criterion values for each matrix $\{J_{\mathcal{P}}^1, \dots, J_{\mathcal{P}}^N$. The index $i = i_u$ such as $\max_{i=1,\dots,N}(\{O(J_{\mathcal{P}}^1), \dots, O(J_{\mathcal{P}}^N))$ permits us to identify the poses $\zeta^- = \zeta^{i_u}$, which have to be removed.

The process is repeated until $\zeta^+ = \zeta^-$.

6. Results

We use the implicit calibration method to identify the kinematic parameters of INRIA's Left-hand Robot given in Table 1 (see Merlet 2000). The leg length L_i, j varies between 0 and 3.5 cm.

6.1. Simulation

As a preparation, we apply the algorithm (Section 4.1) to the Jacobian matrix (5) to find poses that minimize the effect of measurement noise. First we use the IOOPS (see Section 4.1) to determine the measurement poses which maximize the value of the criterion O_i, $i = 1, \dots, 4$ presented in Section 2. The parameters of our algorithm are given as follows.

- The initial value of Υ^N to initialize the Jacobian values is chosen randomly inside the MW, which is bounded by the constraint described in Section 5.3.
- The initial value of ζ_U is chosen as the same process inside the MW. These $i = 1, \dots, U$ poses are used to initialize the U successive executions of the procedure AddFind.

The values of O_i, $i = 1, \dots, 4$ given by maximizing the criterion O_1 are given by Figure 6 as the function of the number of IOOPS iteration (number of selected poses $N = 18$). The four curves correspond to each criterion O_i, $i = 1, \dots, 4$. Note that the process stops after a few iterations, which is approximatively equal to the number of measurements used to calibrate the robot. As predicted, the observability indices associated with the new set of configurations have been increased relative to the old values for all O_i, $i = 1, \dots, 4$. Furthermore, the optimal poses obtained from the algorithm (Section 4.1, Figure 6) for the criterion O_1 have the desirable property of tending to converge to optimal poses on the boundary of the articular workspace, where leg lengths are minimal or maximal. Figure 8 presents these poses in term of their articular coordinates. In each case, we observe an undesirable tendency among certain configurations to converge at poses near the initial ones, which we attribute to the presence of many local maxima. The values obtained for each criterion are lower than the results proved by the experimentation given in Figure 6. To choose the criterion O_1 for improving $J_{\mathcal{P}}$ gives the best measurement poses in term of maximization of all criterion O_i. We now repeat the above procedure with O_2, O_3 and O_4, respectively, in place of O_1 (see Figure 7).

We want to observe the localization of poses obtained with IOOPS maximizing the criterion O_1 and given at the end of the process illustrated by Figure 6. Figure 8 plots the values of the 18 × 6 leg length before and after computing IOOPS. We observe that the selected poses are localized at the boundary of the calibration workspace. This remark is important: the natural way to explain this observation is that these configurations

Table 1. Theoretical Values of the Kinematic Parameters of INRIA's Left-hand Robot in cm

Leg	a_x	a_y	a_z	b_x	b_y	b_z	l
1	−9.7	9.1	0	−3	7.3	0	52.2496
2	9.7	9.1	0	3	7.3	0	52.2496
3	12.76	3.9	0	7.822	−1.052	0	52.2568
4	3	−13	0	4.822	−6.248	0	52.2568
5	−3	−13	0	−4.822	−6.248	0	52.2568
6	−12.76	3.9	0	−7.822	−1.052	0	52.2568

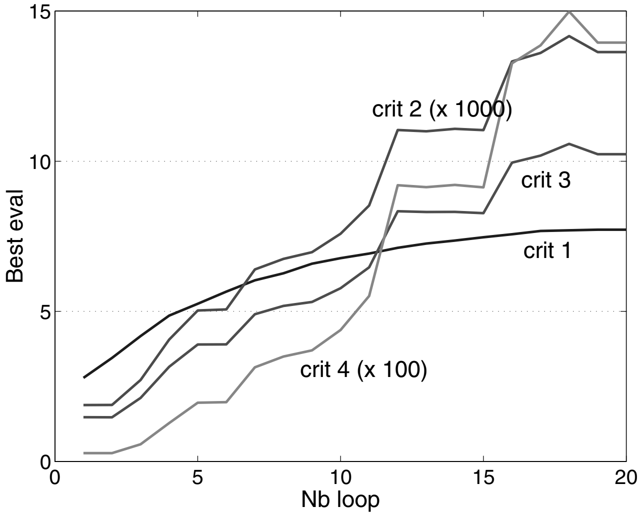


Fig. 6. The values of the criteria O_1, O_2, O_3, O_4 as a function of the iteration number of IOOPS. Here, the chosen optimized criterion is O_1, the number of poses is $N = 18$, and the number of executions of AddFind is $U = 40$.

stimulate at most the variation of pose localization. However, it is interesting to introduce a specific set of post-selected configurations. For this purpose, we begin by constructing the set of $2^6 = 64$ articular configurations corresponding to all possible poses with legs of minimal or maximal length. For each of these, we then use forward kinematics using interval analysis to calculate all corresponding poses defined in terms of generalized coordinates. This problem serves as a benchmark in Merlet (2004). After eliminating unrealistic configurations and applying the constraint on the workspace introduced in Section 5.3, we are left with 44 configurations of measurement expressed in terms of generalized coordinates. The last line of Table 2 gives the value of the observability indices.

We test the influences of the initial configurations for Υ^N (initializing IOOPS) and ζ_U (initializing AddFind) on algorithm convergence. We chose to study three cases:

- random poses inside the MW (denoted RR);
- uniformly distributed poses all over the MW (denoted UU);
- poses localized at the boundary of the MW (denoted BB).

The results are presented in Figure 9. Whatever the choice of the index, the best results are obtained by choosing the initial poses on the boundary of the workspace.

We conclude that the IOOPS algorithm may be parametrized such that:

- if the index O_1 is chosen as the criterion, the results are good for all cases of initialization;
- if one of the indices O_2, O_3 or O_4 is chosen as the criterion, the results are good only by an initialization for measurement poses localized at the boundary of the workspace.

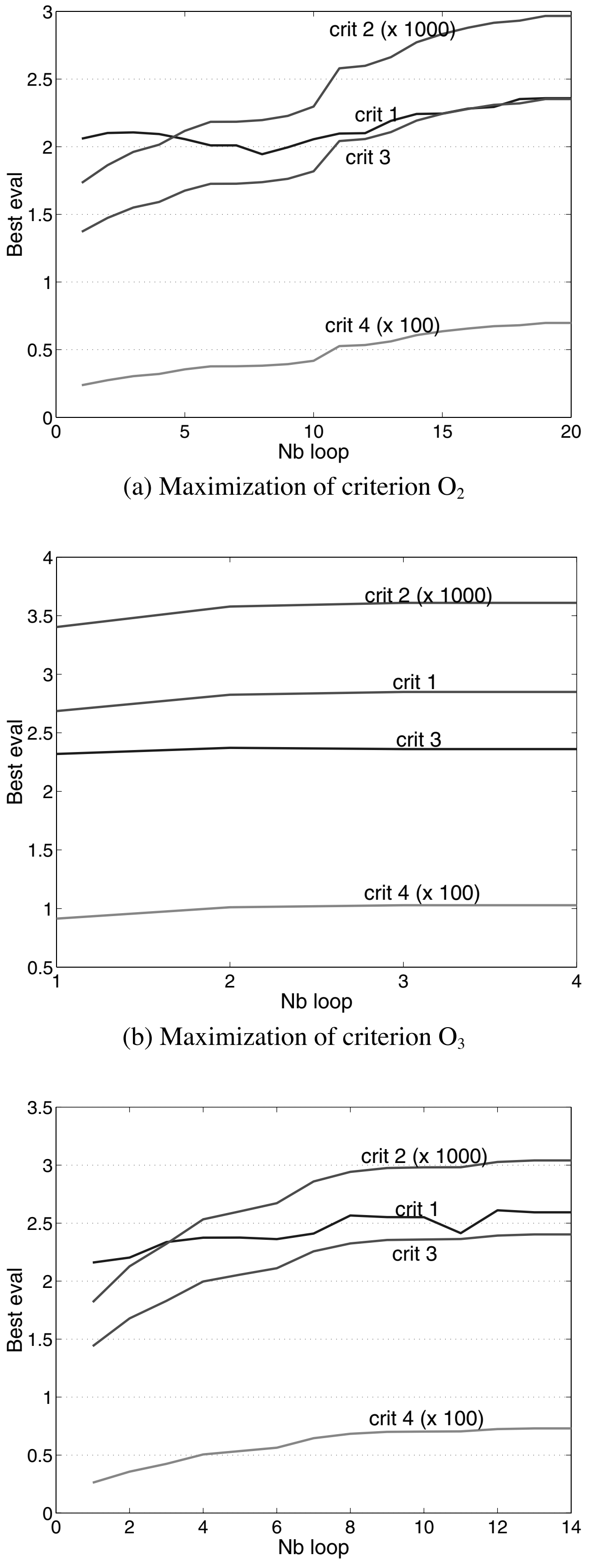
These parametrizations are adequate for the presented case of the calibration procedure. However, it is easy to isolate the boundary workspace poses as an interesting set of solutions. In general, the localization of a set of significant poses may be difficult to obtain. We then prefer to improve our algorithm to find a set of poses that maximize several criteria for an initial estimation of poses chosen randomly.

6.2. Multicriteria Improvement

The multicriteria optimization is a difficult problem. We decided not to define a weighted criterion mixing the criteria presented in Section 2.

An observation of the curves in Figures 6 and 7 shows that the criteria O_2, O_3 and $O_4 = O_2 \times O_3$ should be linked. As a result, we chose to restrict ourselves to the study of a primary O_1 and a secondary O_4 criterion.

The first tested improvement consists of switching the criteria that have to be maximized during IOOPS processing. The criterion is chosen alternatively between O_1 and O_4 with a given probability. The result obtained in Figure 10 shows the poor improvement obtained with this procedure compared to Figure 6. In fact, a relatively small variation of O_1, provided by a particular selection of poses, may change measurably the value of O_4. Then the effect of the randomization is null.

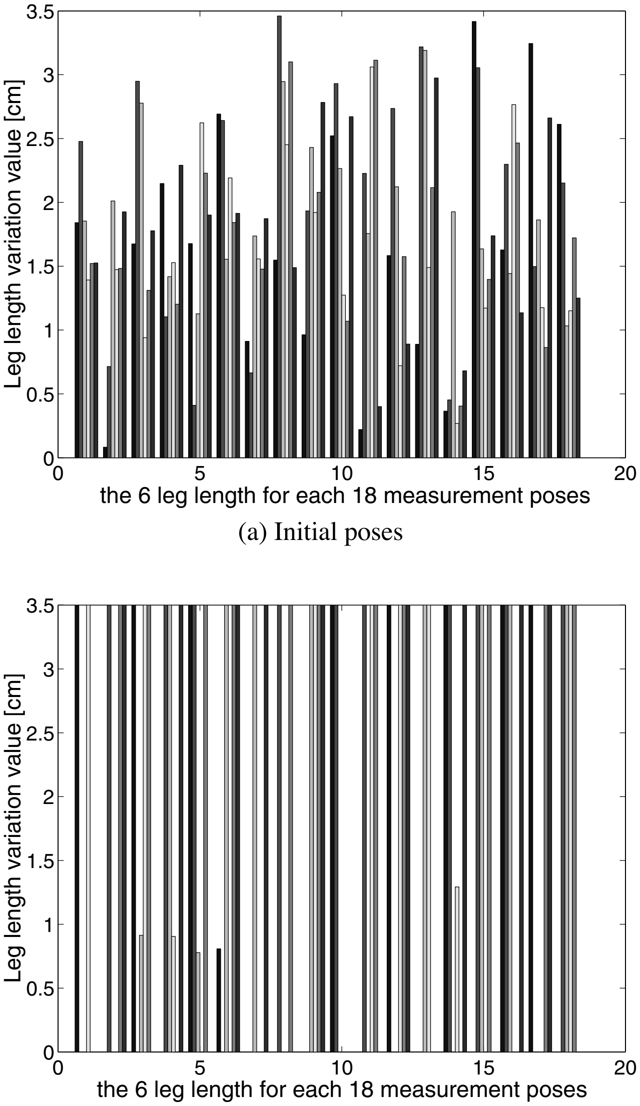


(a) Maximization of criterion O$_2$

(b) Maximization of criterion O$_3$

(c) Maximization of criterion O$_4$

Fig. 7. The IOOPS used the criterion O_2, O_3 or O_4 to select optimal measurement calibration poses.



(a) Initial poses

(b) Selected poses by IOOPS

Fig. 8. Leg length of $N = 18$ poses before and after processing the IOOPS algorithm.

As regards this remark, we propose a second tested improvement. The procedure AddFind is repeated U times and provides U candidate poses (see Figure 2) which improve one criterion (named the primary criterion). For the basic version of AddFind, only the pose, which maximized at most the primary criterion, is selected. The proposed improvement is to select the pose with regards to the secondary criterion in place of the primary. In practice, we do not keep all U candidates for the selection list, but only a percentage (80%) of the best on the primary criterion. Figure 11 shows that improvement provides better values of O_2, O_3, O_4 for the approximatively equivalent value of O_1.

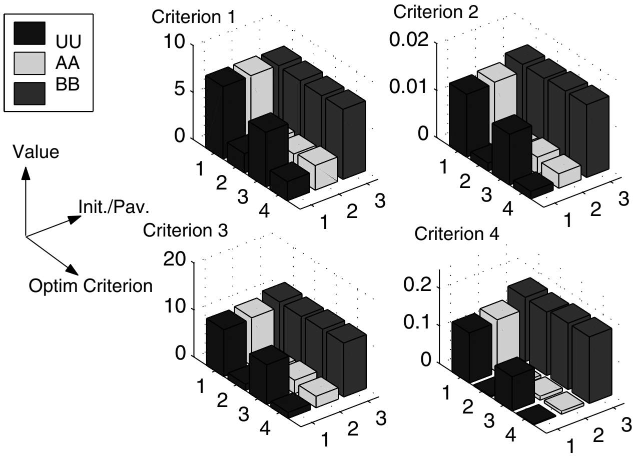


Fig. 9. Criteria value as a function of the localization of initial poses ($n = 18$).

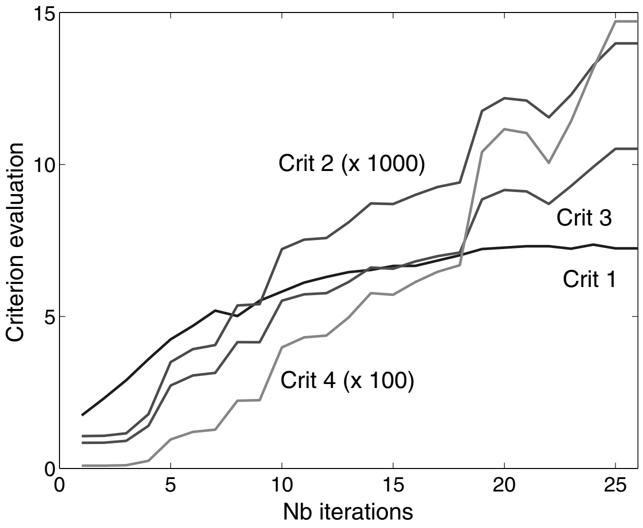


Fig. 10. Alternative random maximization of the criteria O_1 and 0_4.

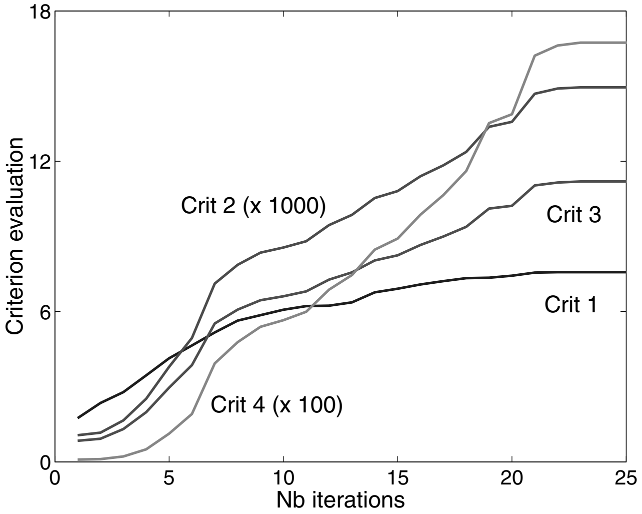


Fig. 11. Maximization of the criterion O_1 but the chosen configuration with criterion O_4.

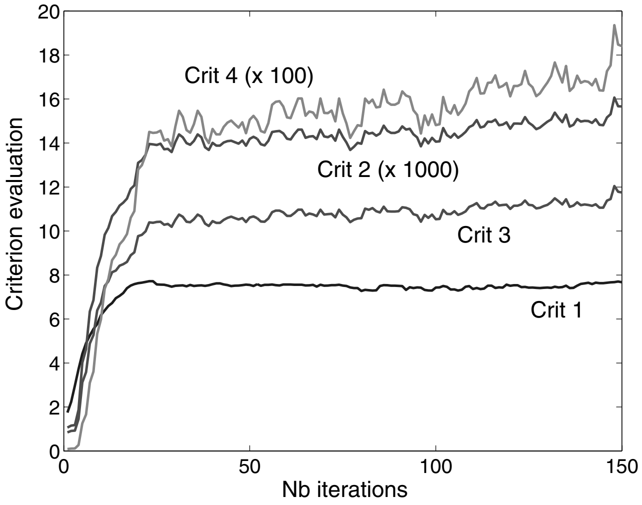


Fig. 12. Maximization of the criterion O_1, but the chosen configuration with criterion O_4 with a size 4 Tabu method.

6.3. Tabu Improvement

Figure 12 clearly shows the evolution of the best solution during the search. There is alternatively degradation and improvement of the best solution during the run until best solutions are found. The solutions found by the Tabu search by using the best tuning for the criterion are the best solutions we have found.

7. Validation

We want to validate the influences of the selected optimal poses on the calibration process. Only calibration simulations permit us to compare the values of the kinematic parameters, obtained after a calibration method with or without a selection of measurement poses, with the actual ones. Then, we use a simulation of the implicit method described in Section 5.2.

Here, we represent the robot design given by constructor in terms of the 42 kinematic parameters described in Table 1. These are denoted by $\mathcal{P}_e$, and serve as initial estimates. They are then perturbed by 0.5 cm, which is even greater than a perturbation simulating a realistic robot design. Perturbated parameters are denoted by $\mathcal{P}$. We now substitute these values in eq. (3) to determine the articular coordinates L_i as functions of four sets of poses, Υ_r, Υ_{s_1}, Υ_{s_4} and Υ_b, expressed in terms of generalized coordinates.

Table 2. Best Value of Observability Indices Extracted from the Experimentation (Figure 12)

	O_1	O_2	O_3	0_4
Υ_r	1.7438	0.0011	0.8431	0.0009
Υ_{s_1}	7.7177	0.0139	10.3760	0.1446
Υ_{s_4}	7.6742	0.0161	12.0471	0.1936
Υ_b	7.5388	0.0151	11.1560	0.1687

The three first sets of poses are extracted from the experimentation presented in Figure 12, as follows.

- Υ_r consists of 18 randomly generated poses inside the calibration workspace. This set is used to initialize the algorithm IOOPS applied with a tabu and a mixed choice on the optimized criterion (see Section 6.3). It corresponds to iteration 0 of Figure 12.

- Υ_{s_1} consists of 18 poses that maximize the criterion O_1. This set of poses corresponds to iteration 24 of Figure 12.

- Υ_{s_4} consists of 18 poses that maximize the criterion O_4 (but also O_2 and O_3). This set of poses corresponds to iteration 148 of Figure 12 and is represented in Figure 13.

The last set of poses Υ_b consists of 18 configurations chosen on the boundary of the workspace for minimal and/or maximal leg length. These are computed using an exact forward kinematic (see Section 6.1). Table 2 gives the value of each criterion.

To each of these sets of poses we now add a uniformly distributed error on measurement parameters to simulate realistic sensor noise. The amplitude of error on our measurement parameters is 0.001 cm for position, 0.01° for the Euler angles defining orientation, and 0.001 cm for leg length. These amplitude noises are applied in all presented simulations.

Next we apply four different simulations of the implicit method to each leg $i = 1, \ldots, 6$. In the first, we use Υ_r to obtain $6 \times 7 = 42$ kinematic parameters, which we denote by $\mathcal{P}_r$.

In the same way, we use Υ_{s_1}, Υ_{s_4}, and Υ_b to obtain $\mathcal{P}_{s_1}$, $\mathcal{P}_{s_4}$, and $\mathcal{P}_b$.

To avoid the problem of a particular direction of the error on measurement parameter, we repeat 100 times the four calibration simulations. Discrepancies between the mean of estimated kinematic parameters and their actual values for our 42 kinematic parameters are listed in Figure 14.

The error $|\mathcal{P}_e - \mathcal{P}|$ obtained before calibration is in column 1, and the remaining columns give the errors associated with poses obtained as a result of our calibration procedure. The mean of post-calibration error $|\mathcal{P}_r - \mathcal{P}|$ associated with 18 randomly chosen poses is in column 2, while the error $|\mathcal{P}_{s_4} - \mathcal{P}|$ associated with 18 poses selected (Υ_{s_4}) as in Section 6 is in column 3. The table in Figure 14 presents also the mean and standard deviation errors of $|\mathcal{P}_{s_1} - \mathcal{P}|$ and $|\mathcal{P}_b - \mathcal{P}|$. As we see in Figure 14, selecting poses as in Section 6 leads to kinematic parameter estimates whose errors are 10–15 times as small as those obtained from randomly chosen configurations.

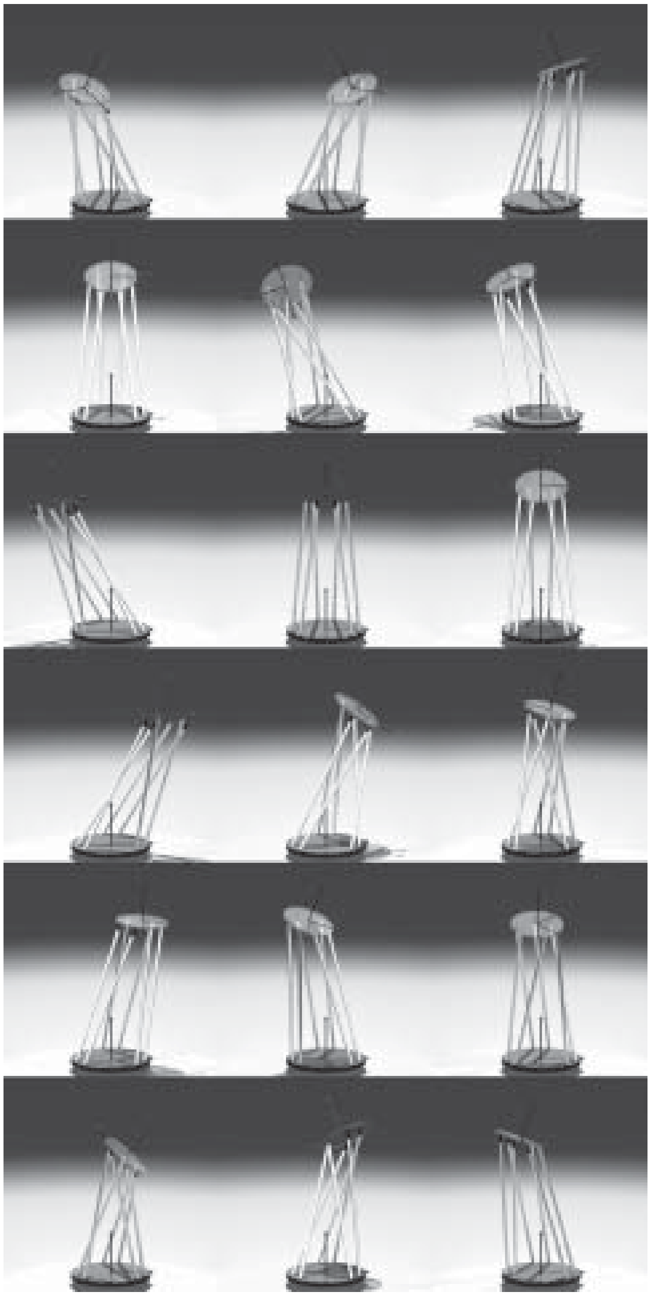
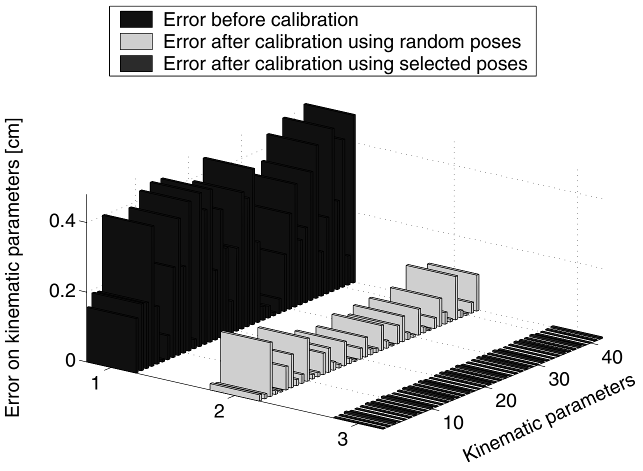


Fig. 13. Representation of the 18 poses of the Gough platform from the set Υ_{s_4}.



(a)

Error on kinematic parameters	Before calibration	After calibration using random poses
Mean [cm]	0.2512	0.0402
Standard deviation [cm]	0.1469	0.0383

(b) Error after calibration obtained by using selected measurement poses

Error on kinematic parameters	After calibration using		
	selected poses		Boundary
	best (O_1)	best (O_4)	poses
Mean [cm]	0.0026	0.0024	0.0027
Standard deviation [cm]	0.0018	0.0013	0.0018

(c) Error before calibration and after obtained by using random measurement poses

Fig. 14. Comparison of the error (mean and standard deviation) on kinematic parameters before and after calibration for selected versus random $N = 18$ measurement poses. Magnitude of the error on kinematic parameters estimation is 0.5 cm, on position/leg lengths measurement, 0.001 cm, and on orientation measurement, 0.01°.

In practice, increasing the number of poses until a certain threshold may improve the robustness of calibration with respect measurement noise, but it increases the time needed for obtaining data necessary for calibration (Nahvi, Hollerbach, and Hayward 1994). To illustrate the above phenomenon, we simulate $41 - 8$ successive calibrations under the same experimental conditions presented at the beginning of the section, while varying the number of measurement poses between 8 and 41. Figure 15 compares mean errors for kinematic parameter (on a, b, and l) estimates obtained using configurations lying on the boundary of the workspace and those randomly chosen.

Observe that, if the relative errors computed using randomly chosen or selected poses decrease steadily as a function of the number of configurations, the absolute errors obtained from a calibration method using carefully chosen configurations are very small, and hence roughly constant for configurations of 18 or more poses.

We may conclude that choosing configurations along the boundary of our workspace permits a robust calibration of the Gough platform. Using a few carefully chosen poses is more effective for calibration than using many random poses. Moreover, our claims have been tested for a wide variety of Gough platforms, with the same positive results.

8. Experimentation

The IOOPS method has been used in an experimentation of calibration of a DeltaLab "Table of Stewart" robot. This robot is a Gough platform, with a basis of radius 270 mm, a mobile of radius 195 mm, and six legs of lengths varying between 345 and 485 mm. This robot has been designed for academic purposes with not as good precision and accuracy as an industrial robot should have.

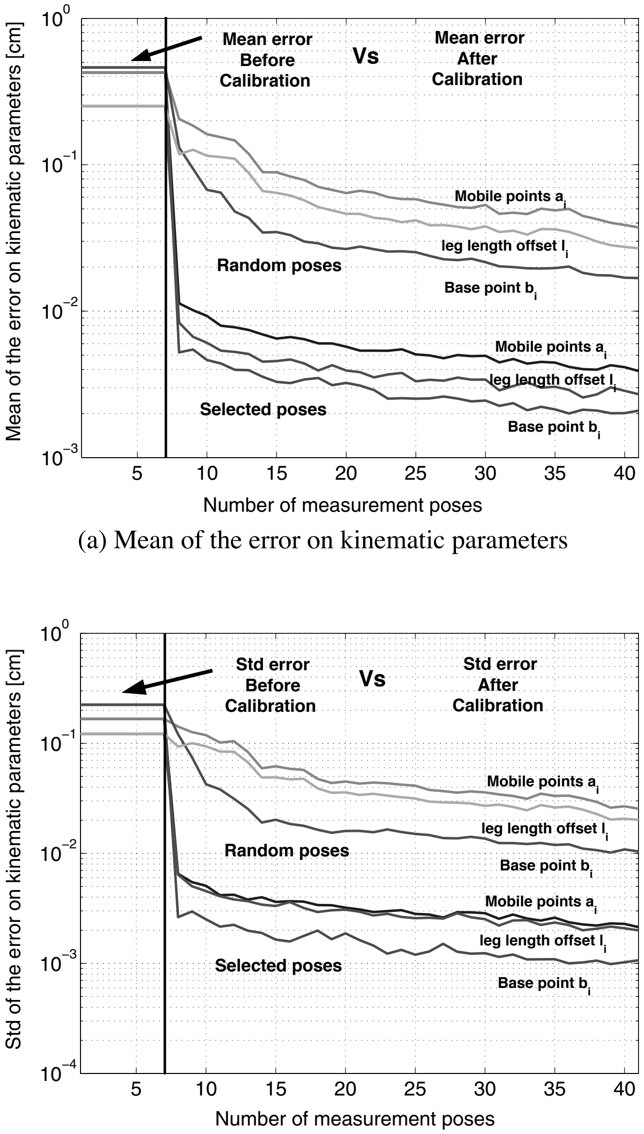
The robot we are working with is located at LASMEA, which is the robotic laboratory of University Blaise Pascal in Clermont-Ferrand, France (see http://wwwlasmea.univ-bpclermont.fr). Practical experimentations have been conducted by N. Andreff and J. M. Lavest.

In order to measure the position/orientation of the end-effector, we are using a vision system (see Figure 16) which consists of

- a digital video camera SONY (1024 × 768) with a 4.2 mm focal;
- an asymmetric target.

This system, once calibrated, produces fully automatically the position/orientation measurements.

An adaptation of the IOOPS method described above has been used to select 13 measurement configurations optimizing the O_4 criterion, starting from a set of configurations located at the boundary of the workspace. We denote by Υ_{ES} this set of poses. Values of the criterion are given in Table 3.



(a) Mean of the error on kinematic parameters

(b) Standard deviation of the error on kinematic parameters

Fig. 15. Comparison of the error (mean and standard deviation) on kinematic parameters before and after calibration for selected versus random measurement poses. Magnitude of the error on kinematic parameters estimation is 0.5 cm, on position/leg lengths measurement, 0.001 cm, and on orientation measurement, 0.01°.

Table 3. Observability Indices for the 13 Selected Poses

	O_1	O_2	O_3	0_4
Υ_{ES}	43.6805	0.0092	47.04	0.4332

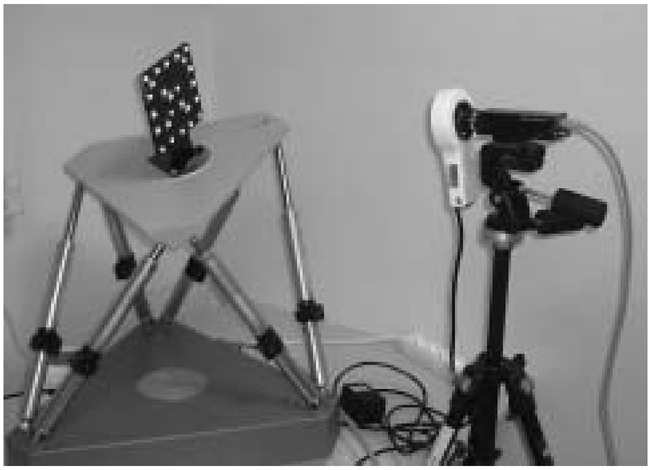


Fig. 16. Table of Stewart (DeltaLab) with vision system.

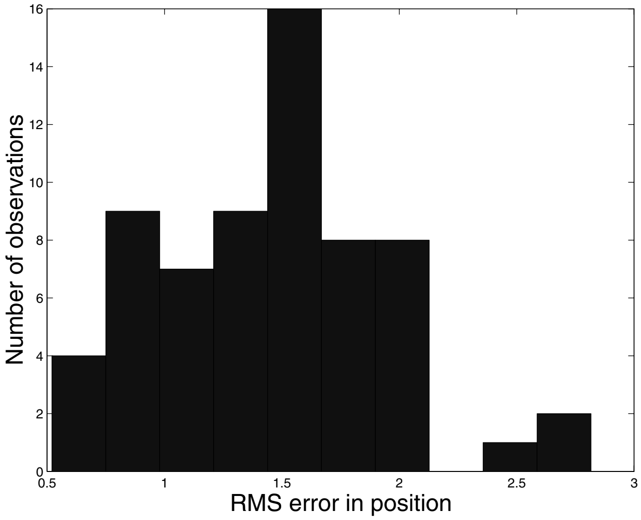


Fig. 17. Error in position for the 64 randomly selected poses, computed for $\mathcal{P}_{ES}$.

Calibration of the robot is then performed by using the implicit calibration method proposed in Section 5.2. This produces the kinematic parameters denoted by $\mathcal{P}_{ES}$.

In order to experimentally validate these results of the calibration, we measure the position and the orientation of the robot in 64 different totally randomly chosen poses (we denote by Υ_{ER} the corresponding set). For each of these poses, we compute the norm of the difference between this measurement and the position/orientation computed by the forward kinematic using $\mathcal{P}_{ES}$ parameters and the measures of leg lengths. The root mean square (RMS) error of position is shown in Figure 17.

Table 4 gives means and standard deviations.

Table 4. Error in Position and in Orientation for the 64 Randomly Selected Poses, Computed for $\mathcal{P}_{ES}$

Error	Position (mm)	Orientation (°)
Mean	1.45	0.27
Standard deviation	0.48	0.27

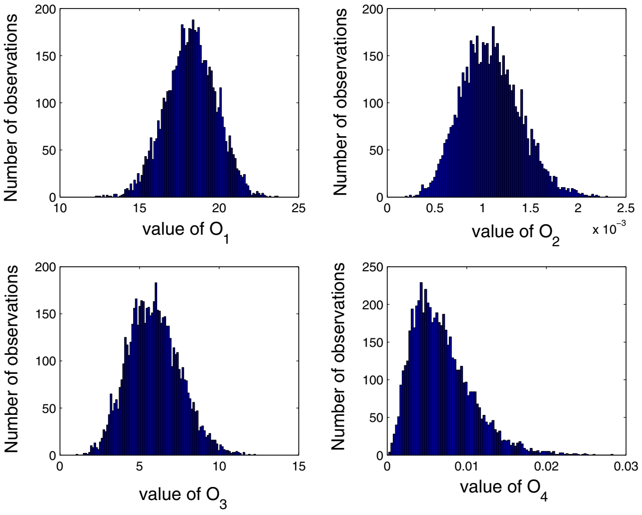


Fig. 18. Observability indices for 6000 times randomly selected poses.

The next step in experimentation consists of comparing the observability indices computed for IOOPS selected poses with the indices obtained for randomly selected poses. For this purpose, we select randomly 13 poses amongst the 64 poses of Υ_{ER}, and we calibrate the robot with these poses to obtain the kinematic parameters $\mathcal{P}_{ER_i}$. This process is repeated 6000 times. For each of the 6000 observations, we compute the indices of observability (see Figure 18). Comparison with the values computed with the Υ_{ES} poses (Table 3) shows how interesting it is to select the poses.

The last step of experimentation is the validation. For each of the 6000 observations, we compute the RMS errors in position and orientation for the 51 (64 − 13) non-selected configurations. To achieve this, we use a forward kinematic computed as a function of the kinematic parameters $\mathcal{P}_{ER_i}$ and the leg length measurements. Errors in position are given in Figure 19, and means and standard deviations are given in Table 5. The results show again the benefit of selecting poses. However, we were expecting even better results on the mean of position errors of Table 4, but investigations show a biased error of 1.29 mm on the z-axis, which is the focal axis of the camera.

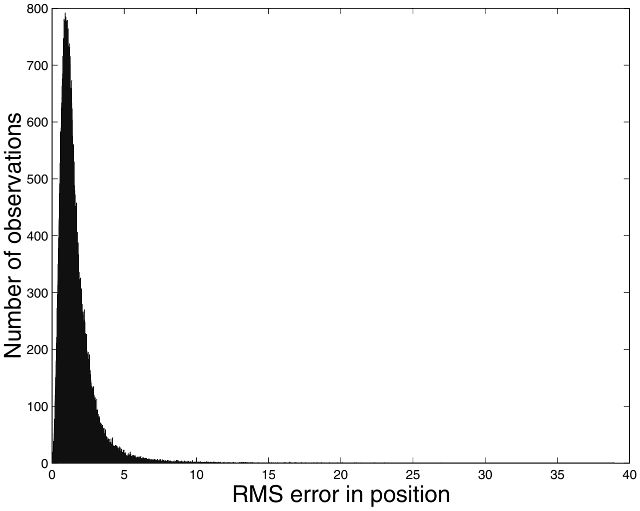


Fig. 19. Error in position for the 51 × 6000 randomly selected poses.

Table 5. Error in Position and in Orientation for the 64 Randomly Selected Poses, Computed for $\mathcal{P}_{ES}$

Error	Position (mm)	Orientation (°)
Mean	1.55	1.66
Standard deviation	1.14	0.96

The IOOPS method has been also used to design a calibration experimentation in an industrial context. Results are presented in Daney (2003).

9. Conclusion

In this paper, we have presented a method to select the calibration poses inside a finite set of measurement data or inside a continuous area describing a measurement workspace. This algorithm is based on a local optimization of an observability index. Additionally, meta-heuristic methods permit us to take into account local minima and multicriteria problems.

Several results may be emphasized, as follows.

- Our results show that by choosing measurement configurations that maximize an observability index, we may significantly increase the robustness of Gough platform calibration with respect to measurement noise. Note, however, that maximization with respect to different indices related to our identification Jacobian has varying effects on calibration and that the appropriate choice of index depends on the effects intended, as expressed in Nahvi and Hollerbach (1996).

However, we claim that the choice of this index O_1 (the product of singular values) is best suited to our algorithm, because of the following.

  - The optimization process is less sensible to local minima by using this index (O_1) in place of O_2, O_3 or O_4 (condition number, smallest singular value or the product of these).
  - All observability indices (O_1, O_2, O_3 or O_4) are maximized or close to being maximized iff the optimized criterion is O_1.

- We can observe that the optimal poses are localized near the boundary of the workspace. We do not have a formal proof of this observation. However, intuitively, it is easy to understand that a wide stimulation (in contrast to local) of a system permits its good observation. This remark is important: to choose the poses on the boundary provides some interesting approximation of optimal poses, or a good initial configuration for an algorithm of experimental design.
- The presented simulation shows the interest of the selection of calibration poses on the identification results. The error on kinematic parameters due to measurement noise is largely reduced (divided by 10–15). Moreover, it is not necessary to multiply the number of measurement poses. We observe that a number of poses greater than 20 does not improve meaningfully the robustness of the identification. In comparison, this threshold is around 40–50, if the poses are not selected by an algorithm. Decreasing the number of measurements is an interesting result for the practical case.

For future work, it would be interesting to explore further the relationship between the geometry of the poses obtained by our algorithm and the numerical properties of our constraint equations. Among the most intriguing properties of our poses are that they lie on the boundary of our articular workspace and that they are highly radially symmetric. It seems natural to conjecture that the symmetry of our Gough platform and the configurations we obtained are linked; more optimistically, we might hope to reduce the number of kinematic parameters needed to define a set of poses optimal for calibration.

Acknowledgments

The author would like to acknowledge the members of the Coprin team for their various suggestions and their help.

References

Baeck, T., Fogel, D. B., and Michalewicz, Z. editors. 1997. *Handbook of Evolutionary Computation*. Institute of Physics Publishing, Bristol, UK.

Borm, J. H., and Menq, C. H. 1991. Determination of optimal measurement configurations for robot calibration based on observabililty measure. *Journal of Robotic Systems* 10(1):51–63.

Chiu, Y. J., and Perng, M. H. 2004. Self-calibration of a general hexapod manipulator with enhanced precision in 5-dof motions. *Mechanism and Machine Theory* 39:1–23.

Daney, D. 2002. Optimal measurement configurations for Gough platform calibration. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 147–152.

Daney, D. 2003. Kinematic calibration of the Gough platform. *Robotica* 21(6):677–690.

Driels, M. R., and Pathre, U. S. 1990. Significance of observation strategy on the design of robot calibration experiments. *Journal of Robotic Systems* 7(2):197–223.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operation Research* 13:533–549.

Glover, F., and Laguna, M. 1997. *Tabu Search*. Kluwer Academic, Boston, MA, pp. 408.

Hao, J. K., Galinier, P., and Habib, M. 1999. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'Intelligence Artificielle* 13(2):283–324.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 1st edition. University of Michigan, MI.

Khalil, W., Gautier, M., and Enguehard, Ch. 1991. Identifiable parameters and optimum configurations for robots calibration. *Robotica* 9:63–70.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* 220(4598):671–680.

Lemaréchal, C., Bonnans, J. F., Gilbert, J. C., and Sagastizabal, C. A. 2003. *Numerical Optimization*. Springer-Verlag, Berlin.

Lintott, A., and Dunlop, G. 1996. Calibration of a parallel topology robot. *Proceedings of the 6th International Symposium on Robotics and Manufacturing (ISRAM)*, Montpellier, France, pp. 429–434.

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

Merlet, J.-P. 2000. *Parallel Robots*. Kluwer, Dordrecht.

Merlet, J.-P. 2004. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *International Journal of Robotics Research* 23(3):221–236.

Mitchell, T. J. 1974. An algorithm for the construction of *d-optimal* experimental designs. *Techometrics* 16:203–210.

Nahvi, A., and Hollerbach, J. M. 1996. The noise amplification index for optimal pose selection in robot calibration. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, MN, pp. 647–654.

Nahvi, A., Hollerbach, J. M., and Hayward, V. 1994. Calibration of a parallel robot using multiple kinematics closed loops. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, CA, pp. 407–412.

Rouillier, F., Faugère, J. C., Merlet, J. P., and Rolland, L. 2005. Forward position analysis for parallel robots. In preparation.

Takeda, Y., Shen, G., and Funabashi, H. 2004. A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series. *Transactions of the ASME* 126:856–865.

Vischer, P. 1996. Improve the accuracy of parallel robots. Ph.D. Thesis, École polytechnique fédérale de Lausanne.

Walter, E., and Pronzato, L. 1997. *Identification of Parametric Models*. Springer-Verlag, Berlin.

Zhuang, H., Wang, K., and Roth, Z. S. 1994. Optimal selection of measurement configurations for robot calibration using simulated annealing. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, CA, pp. 393–398.

Zhuang, H., Wu, J., and Huang, W. 1996. Optimal planning of robot calibration experiments by genetic algorithms. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, MN, pp. 981–986.

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