Robotica (2003) volume 21, pp. 677–690. © 2003 Cambridge University Press
DOI: 10.1017/S0263574703005083 Printed in the United Kingdom

Kinematic calibration of the Gough Platform

David Daney

INRIA, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
E-mail: David.Daney@sophia.inria.fr

(Received in Final Form: February 11, 2003)

SUMMARY
Kinematic calibration is essential to improve the accuracy of the manipulator. This paper presents a complete description of the Gough platform modeling and a unified scheme to identify its kinematic parameters. The interest of this formulation is that it may be applied whatever information is available on the state of the robot (measurement or constraints) without using the kinematics to obtain the basic system of constraint equations. Moreover, the scheme may be applied for all parallel robots. We propose to experiment and compare three methods of calibration, using either (or both) external measurement and internal redundant sensors. Finally, we show how to reduce the initial error in pose *determination* by 99% for the Hexapode 300, CMW's machining center, and validate the choice of a self-calibration method in an industrial context.

KEYWORDS: Parallel robot; Kinematic calibration; Optimization; Gough platform.

1. INTRODUCTION

Parallel robots, in particular the Gough/Stewart platform, offer high accuracy for manipulating heavy objects, even at high velocities, hence their use in a large array of industrial applications, from flight simulators to medical applications such as surgery and more recently for high-speed machining centers.

It is well-known that the geometry of robotic manipulators does not exactly match the design goals due to defects in manufacturing or assembly. A direct consequence is to reduce the accuracy of the manipulator, since robot control heavily relies on a precise description of the kinematic model. One way to tackle this problem is by improving the theoretical kinematic model by means of kinematic calibration.

Let us employ the paradigm of reference [1] in stating a unified calibration formulation. First, the principle is to link the unknown kinematic parameters P and the information on the state of the robot M, either provided by sensors or through constraints applied on joints or brought by an additional mechanism. Some loop equations $f(P, M) \simeq 0$ can be determined; the equations vanish within the measurement error. The dimension of these constraints for each configuration of the robot supply the dimension of the redundant information concerning the state of the robot.

The simplest way to determine M is by using the internal sensors of the manipulator. Usually, though, as they do not provide redundant information, their number is minimal for controlling the manipulator's degrees of freedom (dof). It is possible to install additional captors on passive joints for self-calibration[2] (with the benefit of simplifying the forward kinematics).[3] Alternatively, one can decrease the dof by constraining the end-effector[4,5] or the mobility of some joints;[5,6] this can also be used to simplify the calibration procedure if external measurements are present.[7–9]

In practice, it is not easy to add redundant sensors or constraints. Hence, most calibration methods use external measurements devices to obtain the required information, such as laser trackers, theodolites,[10] cameras, inclinometers[11] or mechanical devices.

Many authors use the kinematics to relate the kinematic parameters P to the available information M. Then, the basic calibration methods with external measurements use either the forward kinematics[12,13] or the inverse kinematics.[14] Those calibration methods may be prone to error. The reason is the difficulty to obtain a closed form for the solutions of the kinematics problem. Wampler et al. in reference [15] introduce *Implicit Calibration*. They present a unified formulation which is independant of the type of information used to identify the kinematic parameters. The principle is to consider the basic system of equations induced by the closed loop nature of parallel robots. The system is then specified as a function of the available data. Then the free joint variables, unconstrained or unmeasurable, are eliminated from the basic system to provide the constraint equations. Solving this system allows the kinematic parameters to be identified. Vischer[16] extends the method on Delta and Argos robot and commends a direct use of the Implicit equations. In many cases, the kinematics models are used to numerically avoid the unwanted joint variables. But this kind of method may be problematic in term of existence and unicity of its solution, within a calibration optimization loop. We prefer to use an algebraic elimination method which improves the result of the identification in term of noise measurement robustness.[17]

First, the modeling of a Gough platform is reviewed in details. Then, a formulation of the *Implicit method* is applied on the Gough platform. This allows a unification of any type of calibration methods which may be adapted on any type of closed-loop manipulator. This is illustrated by three different methods of calibration which are obtained for the same process. Then, we present the experimentation and the comparison of three methods applied for the kinematic calibration of a high speed machining center. The interest of the self-calibration method compared to the methods using an external measurement device is then discussed.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

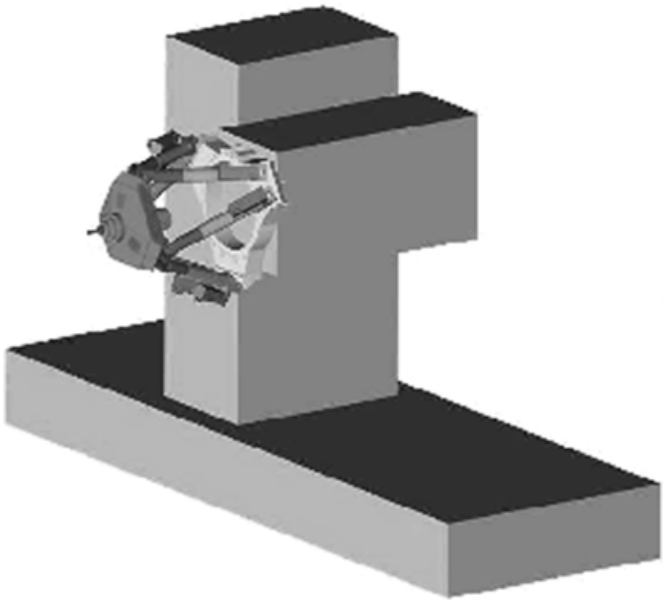


Fig. 1. CMW's milling machine 9× concept.

2. PROBLEM STATEMENT

C.M.W. (Constructions Mecaniques des Vosges) proposes to improve a classical machining center 3 axes by adding, in place of the tool, a mechanism to provide a High Speed Machining with 5 dof (a 6th dof is used to increase the efficiency of the mechanism – see reference [18]) (Figure 1).

This additional head, the *Hexapode 300* of CMW (see http://www.hexapode.com/), is based on a Gough platform.

One of CMW's problems concerns the customer support in the case of an intervention on the robot. If one leg of the *Hexapode 300* is disassembled, an identification of its kinematic parameters is necessary. The question of the choice between a calibration method with external measurement and a self-calibration with redundant sensors is then very important. In the case of a classical calibration method with external device which measure directly or indirectly the accuracy of the robot, the cost of the experimentation and the external devices is high. In the case of a self-calibration, the cost of the robot itself is increased by adding the redundant sensors, but the calibration procedure is completely automatic, fast and cheap and may also be used to improve the control of the robot by simplifying the direct kinematics calculation. To overrule the question, CMW wants to compare both types of calibrations in terms of improvement of the modeling knowledge. We propose to compare three types of calibration by experimentation.

In addition to the 6 linear sensors installed on the prismatic joints to measure the leg length variations, twelve incremental encoders are installed on the 12 revolute joints of the 6 U-joints of the robot – see 3.1.1, to provide their rotation values. Moreover, a measurement machinery is used to provide the position and the orientation of a frame defined by a target installed on the mobile platform with respect of a reference frame associated to a target installed on the base frame.

With these experimental configurations, it is possible to propose 3 different calibration methods.

- The first method (denoted **M-I**) is the basic *Inverse Method* introduced in reference [14] using only the information of pose and the leg length measurement.
- The second method (denoted **M-II**) is a self-calibration procedure using only the information provided by internal sensors: leg lengths and U-joints measurement values. This method uses the same type of information as the method presented in reference [2], but we will see that the *Implicit method* provides a system of equations close to the constraints introduced in references [19,20].
- The third method (denoted **M-III**) consists in using both types of information available: the external measurement provided by the measurement of the pose of the robot and the internal measurement provided by the leg lengths and U-joint measurements.

We have to note that this work is only a part of the total calibration of the *Hexapode 300*. Firstly, the non-kinematic errors due to temperature variations or backslash in joints are not taken into account in the proposed calibration methods. This is possible because CMW has installed temperature regulation of the machining tool and by a high quality of the joints. Secondly, we propose to identify the internal kinematic parameters of the robot. Positioning the robot with respect of the machining object or the installation of a tool on the end-effector is a task which may be realized independently by classical methods.

3. GOUGH PLATFORM MODEL

We want to determine the type and the minimal number of kinematic parameters of the Gough platform modeling (Figure 2). Masori et al. present a solution[12] with 132 parameters. Vischer[16] proposes a formula in order to calculate this number for multi-loop robot and set it in the case of the Gough platforms and obtained 138 kinematic parameters. These two results agree under the hypothesis presented in reference [12] and underlined in reference [16].

We present our own model to simplify the following development. It is based on the modeling of each part of the chains which compose this multi-loop robot. We have first to study the mechanical structure of the legs (or segments). Then the links between the two rigid bodies, the base and

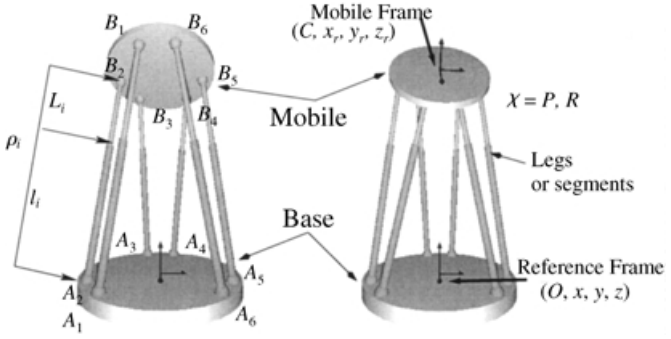


Fig. 2. Gough's platform.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

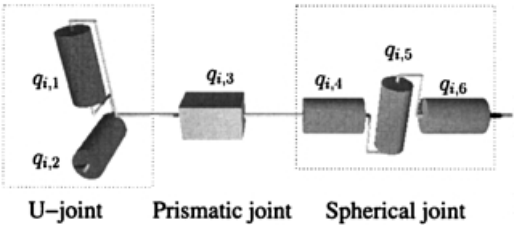


Fig. 3. A 2RP3R chain.

the mobile platforms and legs are studied. Moreover, the kinematic parameters associated with additional calibration system and sensors will be presented.

3.1. The kinematic parameters of the legs

3.1.1. The complete model. Each leg $i=1, \ldots, 6$ of the Gough platform is connected to the base platform by a U-joint, composed of two revolute joints RR, and to the mobile platform by a spherical joint, composed of three revolute joints RRR. Initially, these joints are not instrumented. The mobility of the platform is obtained by controlling the legs length variations using a linear actuator (actuated P joint). Hence, this kinematic chain may be considered as a serial robot of type 2RP3R (see Figure 3). The joint variables composing this chain will be denoted $q_{i,1}$, $q_{i,2}$ for the two first revolute joints (U-joint), $q_{i,3}$ for the prismatic joint and $q_{i,4}$, $q_{i,5}$, $q_{i,6}$ for the last revolute joints (spherical joints).

The parameters which model the 2RP3R kinematic chain may be obtained by using the Denavit-Hartenberg modeling convention[21] adapted for the Gough platform.

- The reference frame of the segment i is noted $\Omega_i^a(A_i, \mathbf{x_i^a}, \mathbf{y_i^a}, \mathbf{z_i^a})$. The center A_i is located on the foot of the common perpendicular between the axes of the first two revolute joints defining the U-joint.
- The $\mathbf{x_i^a}$ axis lie along the axis of the first revolute joint defining the 2RP3R serial chain and $\mathbf{z_i^a}$ along the common perpendicular between the axes of the two first revolute joints defining the U-joint, when the first revolute joint value is set to zero (with respect to some reference that may be provided, for example, by a sensor).
- The end-effector frame of the segment i is noted $\Omega_i^b(B_i, \mathbf{x_i^b}, \mathbf{y_i^b}, \mathbf{z_i^b})$. This center B_i is located on the foot of the common perpendicular between the axes of the last two revolute joints defining the spherical joint.
- The $\mathbf{x_i^b}$ axis lie along the axis of the last revolute joint defining the 2RP3R serial chain and $\mathbf{z_i^b}$ along the common perpendicular between the axes of the last two revolute joints defining the spherical joint, when the last revolute joint value is set to zero.

With the previous convention, 12 kinematic parameters (denoted p_i) are sufficient to define the 2RP3R chain. These parameters have been presented by Masory et al.[12]

3.1.2. Simplifications. Considering all the parameters of the kinematic chain of one robot's leg is difficult in practice. The U-joint or the spherical joint may not be ideal: the distance between joint axes may not be zero, and these axes may not be exactly orthogonal. Moreover, the assembly defects between U-joint, prismatic joint and spherical joint for each legs have to be taken into account. These errors are modeled by the parameters p_i introduced previously.

Masory et al.[12] studied the influence of Gough platform pose errors as a function of 132 kinematic parameters. Under realistic manufacturing errors, they conclude that the influences of defects in the U and the spherical joints have a minor effect on pose accuracy compared to the errors in the location of the U and spherical joint centers and the offset on the leg length. Hence, we can introduce four hypotheses:

H-1 The U-joints are perfect.

H-2 The spherical joints are perfect.

H-3 The prismatic actuators are perfectly assembled with the U-joint.

H-4 The prismatic actuators are perfectly assembled with the spherical joint.

If all these assumptions are satisfied, the parameters p_i are considered as known. The observation of these joint defect parameters is difficult if the values of the passive revolute joints are unknown. Then, this may be integrated in the command of the manipulator. That increases the difficulty to implement the command, and, in fact, does not correspond to the simplified modeling of the Gough platform generally used to control this mechanism without an additional sensor.

In the case study of CMW's Hexapode calibration, the knowledge of the U-joint angular values, provided by additional sensors, permits the identification of U-joint defects. This information is important to improve the future design.

We assume the hypothesis **H-2**, **H-3** and **H-4** are realized. The U-joint defects may be modeled by two kinematic parameters for each leg: the angle (denoted α_i) and the distance (denoted d_i) between the axis of the two revolute joints ($p_i=[\alpha_i, d_i]$). Note: to obtain a value of the defect α_i near zero, the angle between U-joint axes is equal to $\alpha_i+\frac{\pi}{2}$.

3.1.3. The Forward Kinematics of the legs. The Forward Kinematics of the 2RP3R chain provide the position (denoted T_i) and the orientation (denoted R_i) of Ω_i^b with respect to Ω_i^a, as function of the joint variables $q_{i,j}$ and the kinematic parameters $p_i=[\alpha_i, d_i]$ (under hypotheses **H-2**, **H-3** and **H-4**).

We use the homogeneous transformation $H_i=\begin{pmatrix} R_i & T_i \\ 0 & 0 \end{pmatrix}$ to denote the position and the orientation of Ω_i^b with respect to Ω_i^a.

With the previous convention, we obtain:

$$H_i=\mathcal{R}_x(q_{i,1}) \circ \mathcal{R}_z\left(\frac{\pi}{2}+\alpha_i\right) \circ \mathcal{T}_z(d_i) \circ \mathcal{R}_x(q_{i,2}) \circ \mathcal{T}_z(q_{i,3}) \circ \mathcal{R}_x(q_{i,4}) \circ \mathcal{R}_y(q_{i,5}) \circ \mathcal{R}_z(q_{i,6}) \quad (1)$$

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

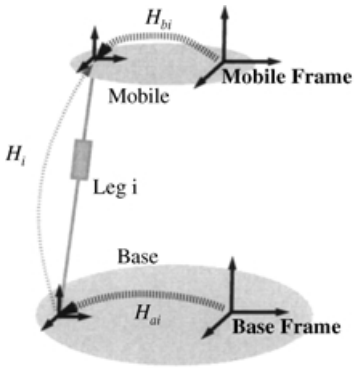


Fig. 4. Closed-loop parameters.

where $\mathcal{R}_u$ (or $\mathcal{T}_u$) denotes the homogeneous transformation of a pure rotation (or translation) around (or along) the axis u of the frame associated with the considered joint $q_{i,j}$ and "$\circ$" denotes the matrix product.

3.2. Closed-loop parameters

(Figure 4) The kinematic parameters which model the loops linking each robot segment i and the mobile and base platforms, have to be identified.

We define two frames, Ω_o associated with the base platform and Ω_c associated with the mobile platform. The vector a_i and the matrix R_{ai} provide the coordinates of the point A_i and the orientation of the frame Ω_i^a with respect to the base frame Ω_o. The vector b_i and the matrix R_{bi} provide the coordinates of the point B_i and the orientation of the frame Ω_i^b with respect to the mobile frame Ω_c.

Then the homogeneous matrix $H_{ai}[R_{ai}, a_i]$ (or $H_{bi}[R_{bi}, b_i]$) characterizes the transformation between Ω_o and Ω_i^a (or Ω_c and Ω_i^b).

We need three independent parameters to model the orientation matrix R_{ai} (or R_{bi}) and three for the position a_i (or b_i). The total number of parameters is so equal to $6\times6\times2=72$

3.3. Calibration additional system parameters

In some calibration methods, it is necessary to introduce an additional kinematic chain to link the mobile and the base platform and therefore to provide a redundant information on the state of the robot. This kinematic chain may be seen as a 7th virtual leg with an index equal to $i=0$ (Figure 5).

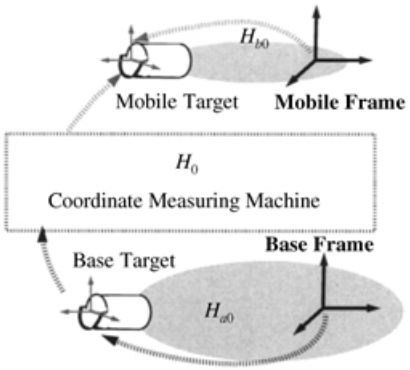


Fig. 5. Calibration system parameters.

We have hence to introduce $6+6=12$ additional parameters to model the transformations between:

- The base frame Ω_o and the reference frame of the 7th leg Ω_0^a.
- The mobile frame Ω_c and the frame associated with the end-effector of the 7th leg Ω_0^b.

By analogy to the 6 legs of the robot, these parameters are denoted a_0 (or b_0) for the position of Ω_0^a (or Ω_0^b) with respect to Ω_o (or Ω_c) and R_{a0} (or R_{b0}) for the rotation matrix defining the orientation of Ω_0^a (or Ω_0^b) with respect to Ω_o (or Ω_c).

Remark: It may not be possible to physically define the center and the axes direction of the frames associated with the additional 7th leg. For example, if a measurement machinery provides only the position of the mobile platform with respect to a base frame, the orientation of Ω_0^b is not known. A physical reference has to be used to fully define this frame. In this case, 3 parameters which define the orientation Ω_0^b have to be fixed.[17,22]

3.4. Sensors parameters

In most cases sensors will provide only relative information on a joint value. Consequently, it is necessary to add a parameter to obtain the absolute information. Typically, for the Gough platform, a linear sensor installed on each prismatic joint measures the length variations of the leg i, denoted L_i. We introduce an additional kinematic parameter, the offset on the leg length denoted l_i, to give the value of the joint $q_{i,3}$ when the sensor value is equal to zero, $L_i=0$, such that:

$$q_{i,3}=L_i+l_i \tag{2}$$

In the same way, if we instrument the U-joint of the robot, some additional parameters have to be introduced. In this case, as presented for the prismatic joint, an offset associated with each revolute joint is defined such that:

$$q_{i,1}=\Theta_i+\theta_i \tag{3}$$

$$q_{i,2}=\Gamma_i+\gamma_i \tag{4}$$

But the design of the instrumentation of the joint can provide several additional parameters, which complicate the calibration procedure. Typically, additional parameters have to be introduced to model the parallelism error between the linear sensor axis and the prismatic axis or to provide the joint variation as a function of the sensor output. A good way to avoid this problem is to calibrate the sensors in the leg, one at a time before assembling the robot.

In the presented *Hexapode 300* calibration case, we use a degree five polynomial to interpolate the value of each instrumented joint as a function of the output provided by sensors. A measurement device has been used to provide the necessary information for obtaining the coefficients of the polynomial.

3.5. The Gough platform modeling

The following section details the kinematic parameters of the Gough platform. First, we present the most general case and then we describe the parameters used for the calibration of the *Hexapode 300*.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

3.5.1. General case. The number of the Gough platform kinematic parameters is:

- $+6\times12$ parameters to describe the kinematic of the 6 2RP3R kinematic chain modeling the 6 legs of the robot.
- $+6$ parameters for the offset associated to the leg length l_i.
- $+6\times6$ ($+6\times6$) parameters to obtain the position a_i (b_i) and the orientation R_{ai} (R_{bi}) parameters to describe the 6 transformations between the base (mobile) frames Ω_o (Ω_c) and the frame associated with the first (last) $\Omega^a{}_i$ ($\Omega^b{}_i$) revolute joint of the 6 segments of the robot.

The number of parameters is thus equal to 150. But this model does not give the *minimal* number of parameters.

The origin's location of the base and the mobile frame is arbitrary. In order to obtain the minimal number of parameters, we have to associate these origins to some reference points on the platform. This simplification allows to fix $6+6$ parameters. For example, we can use the following constraints:

$$a_{1,x}=a_{1,y}=a_{1,z}=a_{2,x}=a_{2,y}=a_{3,z}=0 \quad (5)$$

$$b_{1,x}=b_{1,y}=b_{1,z}=b_{2,x}=b_{2,y}=b_{3,z}=0 \quad (6)$$

In total, if we avoid the problem of additional sensor calibration, we need 138 kinematic parameters (p_i, l_i, a_i, R_{ai}, b_i, R_{bi} with $i=1,\ldots,6$) to model the Gough platform.

3.5.2. The *Hexapode 300* calibration case. Under the hypotheses **H-2**, **H-3** and **H-4**, we decrease the initial number of parameters from 138 to 6×2 parameters (α_i and d_i) to describe only the defects in U-joints and 6×2 parameters to model the offset associated with the U-joint θ_i and γ_i.

Hence we need 102 parameters in total. But, as a function of the type of information used to calibrate the Gough platform, we may avoid determining some of them.

In all of CMW's calibration cases, the 3 parameters modeling the matrix R_{bi} cannot be identified due to the non-instrumentation of the spherical joints. We will see in the following section that those parameters may be eliminated from the implicit equations and do not appear in constraint equations.

Method M-I. In the case of method **M-I**, the U-joint sensors parameters are not considered. Moreover, the available measurements do not provide the necessary information to identify the parameters p_i: thus, we assume the hypotheses **H-1**, **H-2**, **H-3** and **H-4** are satisfied. The 3 parameters which model each matrix R_{ai} are eliminated by the construction of the constraint equations – see Section 4.3. The kinematic parameters are:

- 24 closed-loop parameters a_i, b_i, $i=1,\ldots,6$ without parameters fixed by the Equations (5),
- 6 offsets on the leg length l_i, $i=1,\ldots,6$,
- 12 additional calibration system parameters R_{a0}, R_{b0}, a_0, b_0.

This amounts to a total of **42** parameters.

Method M-II. In the case of method **M-II**, the kinematic parameters a_0, b_0, R_{a0} and R_{b0} are avoided because we do not use the measurement machinery. Under the hypotheses **H-2**, **H-3** and **H-4**, the kinematic parameters are, with the leg index $i=1,\ldots,6$:

- 42 closed-loop parameters R_{ai}, a_i, b_i (the Equations 5 fix 12 parameters),
- 18 offset parameters l_i, θ_i, γ_i,
- 12 parameters to describe the U-joint defects d_i, α_i,

In total, we then need 72 kinematic parameters to model the robot. But as shown in Section 4.4, it is not possible to identify 6 of them; the number is in fact equal to **66**.

Method M-III. In the case of method **M-III**, we have to add the 12 additional calibration system parameters R_{a0}, R_{b0}, a_0, b_0 to the 66 kinematic parameters defined for the method **M-II**. The number of kinematic parameters is then equal to **78**.

4. CALIBRATION SCHEME

The calibration scheme consists of four steps:

- specifying the implicit equations;
- eliminating the unknown joint variables;
- checking the observability of kinematic parameters;
- solving the constraint equations.

Each point will be presented and developed for the *Hexapode 300* calibration.

4.1 Closed-loop equations

Let us consider a parallel robot composed of $\mathfrak{L}$ legs. We may define $\mathfrak{L}-1$ closed-loops linking the joint variables and the kinematic parameters of the robot. Figure 6 shows two loops for a parallel robot where $\mathfrak{L}=3$.

We obtain, for each configuration of the robot, $(\mathfrak{L}-1)\times6$ independent equations (each loop providing 6 independent equations, 3 in position, 3 in orientation), given as:

$$H_{ai}\,H_i\,H_{bi}^{-1}=H_{aj}\,H_j\,H_{bj}^{-1} \text{ with } i,j\in[1,\ldots,\mathfrak{L}],\ i\neq j. \quad (7)$$

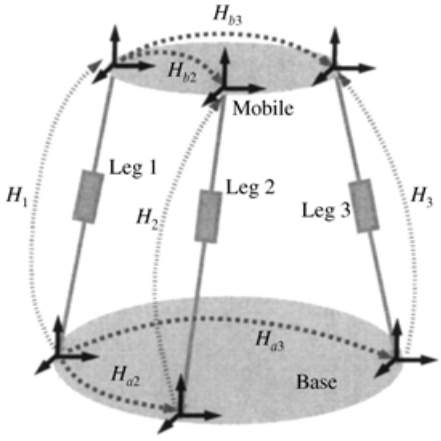


Fig. 6. Two closed-loops.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

In the case of the Gough platform, the number of loops is equal to 5; that provides, for each configuration of the robot, $5\times6=30$ independent equations of type (7).

The matrices H_i – see Section 3.1.3, are function of 5 revolute joints variables ($q_{i,1}$, $q_{i,2}$, $q_{i,4}$, $q_{i,5}$, $q_{i,6}$) with an unknown value for each configuration of the robot and one prismatic joint value ($q_{i,3}$) measured by a sensor. We so get $N_e=30$ Equations (7) for $N_q=30$ unknowns $q_{i,j}$ with $j\in[1, 2, 4, 5, 6]$.

The calibration problem consists in getting redundant information on the state of the robot. In fact, we want the number of equations to be greater than the number of unknowns $N_e>N_q$ for one configuration of the robot.

To do that, there are two simple solutions:

- decrease the number of unknowns; this is the case of the self-calibration problem,
- increase the number of equations; this is the case of the calibration problem.

4.1.1. Self-calibration problem. The self-calibration consists of creating redundant information on the state of the robot such that $N_e>N_q$. For that, the state of the joint variables $q_{i,j}$ has to be either *measured* (a sensor provides the joint value) or *constrained* (the joint is mechanically fixed by a mechanism).

For the Gough platform, if one revolute joint $q_{i,j}$ is *measured* or *constrained*, the dimension of N_q decreases and creates some constraint equations which link the information and the kinematic parameters. The resolution of this system allows the identification of a part (see Section 4.4) or of all kinematic parameters.

4.1.2. Calibration problem. The calibration problem consists in adding an extra kinematic chain (with index $i=0$) linking the base platform to the mobile platform. For instance, an additional mechanism may be installed to *measure* or *fix* partially or totally the dof of the mobile for all configurations of the robot. Let E be the dimension of the constraints created by the extra-chain for each robot configuration. The number of equations is now equal to $(\Re-1)\times6+E$ and provides the necessary redundant information to identify the kinematic parameters. In this case, as presented in Section 3.3, additional kinematic parameters ($\{H_{a0}[R_{a0}, a_0]$ and $H_{b0}[R_{b0}, b_0]\}$) have to be introduced to link the reference frame of this extra-chain Ω_0^a to the base frame Ω_o and the end-effector frame Ω_0^b to the mobile frame Ω_c. The homogeneous transformation of Ω_0^a with respect to Ω_0^b is noted $H_0[R_0, T_0]$.

For the Gough platform, we have to link this 7th segment to the 6 others; we obtain 6 loops of type:

$$H_{a0}\,H_0\,H_{b0}^{-1}=H_{ai}\,H_i\,H_{bi}^{-1} \text{ with } i\in[1, \ldots, 6]. \qquad (8)$$

In general, it is interesting to expand Equation (8). We can divide the loop equations in terms of translation (10) and orientation (9), with i the index of the segment:

$$R_{ai}\,R_i\,R_{bi}^{-1}=R_{aj}\,R_j\,R_{bj}^{-1} \qquad (9)$$

$$R_{ai}\,T_i+a_i-R_{ai}\,R_i\,R_{bi}^{-1}\,b_i=R_{aj}\,T_j+a_j-R_{aj}\,R_j\,R_{bj}^{-1}\,b_j \qquad (10)$$

$$\text{with } i, j\in[0, \ldots, 6] \quad i\neq j$$

4.2. Specification

4.2.1. General cases. The specification consists in fixing the state of each joint variables as a function of the type of the available information.

For a general approach of the problem, the additional kinematic chain providing external measurement or constraining some dof of the mobile platform is modeled as a function of some virtual (or not) joint variables $q_{0,m}$ with $m=[1, \ldots, 6]$ giving an expression of the configuration of this chain H_0. Then, the specification of Equation (10) and (9) may be done as follows: the information on the state of the robot are provided by the values of the joint variables $q_{i,m}$ with $i=[0, \ldots, 6]$, $m=[1, \ldots, 6]$ which may be:

- measured: the value of the joint variable, denoted by $q_{i,m}\rightarrow\widehat{q_{i,m}}$, is measured by a sensor and is known for all robot configuration.
- *constrained*: the value of the joint variable, denoted by $q_{i,m}\rightarrow\overline{q_{i,m}}$, is imposed by a mechanism and does not change with the robot configuration.
- *free*: the value of the joint variable, denoted by $q_{i,m}$, is unknown and varies as a function of the robot configuration.

After the specification of the basic system (8), we obtain, for each configuration of measurement of the robot $K=[1, \ldots, N]$, equations as:

$$F_k(q, \bar{q}, \hat{q}, P)\sim0 \qquad (11)$$

with $q, \bar{q}, \hat{q}$ the set of free, constrained or measured joint variables and P the set of the kinematic parameters.

4.2.2. *Hexapode 300* calibration cases. In the case of *Hexapode 300* calibration, the external measurement device may be considered as an additional kinematic chain providing the position T_0 and the orientation R_0 of Ω_0^a with respect to Ω_0^b. We choose to model this chain as a virtual perfect 2RP3R chain.

The direction of the position (2R) is modeled as a virtual U-joint $q_{0,1}$ and $q_{0,2}$, the norm of the position (P) is modeled as a virtual prismatic joint $q_{0,3}$, the orientation is modeled as a virtual spherical joint $q_{0,4}$, $q_{0,5}$ and $q_{0,6}$.

The virtual axes of joint $q_{0,1}$ and $q_{0,2}$ are defined by the axis x_0^a and x_0^b of Ω_0^a associated with the target installed on the base platform – see Section 5.3. In the same way, the virtual axes of $q_{0,6}$ and $q_{0,5}$ are defined by axes x_0^b and x_0^b of Ω_0^b associated with the target install on the mobile platform.

Table I gives the information available for the proposed methods M-I, M-II and M-III.

Modeling the additional mechanism as a 2RP3R chain does not provide all possibilities of calibration methods. We may use some solutions proposed for serial robot calibration.[4]

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

Table I. State of joint variables, M=measured, C=constrained, F=free.

Using Method	Leg number	Joint variable number 1	2	3	4	5	6
M-I	0	M	M	M	M	M	M
	1, . . . , 6	F	F	M	F	F	F
M-II	0	F	F	F	F	F	F
	1, . . . , 6	M	M	M	F	F	F
M-III	0	M	M	M	M	M	M
	1, . . . , 6	M	M	M	F	F	F

4.3. Elimination

4.3.1. General case. The choice of eliminating variables in a system of equation with uncertainties has to be addressed. Indeed, in general, this procedure decreases the number of unknowns but also increases the degree of the resulting equation. This may indirectly increase the sensitivity of an identification problem to numerical or model errors. Hence, reducing the number of variables may not be the best solution for every kinematic problem.

In the specific calibration problem, we think that the elimination of the free joint variables is a good solution as:

- the number of variables is drastically reduced as otherwise the list of unknowns include the number of the free joint variables multiplied by the number of measurement poses. In that case, identification of all variables, for example, by a least square method will be problematic.
- the degree of the equation obtained after elimination is still small, and consequently the influence on identification is not significant.

The aim is now to eliminate from Equation (11) the free joint variables q in order to obtain a new system of equation named *constraints equations* and denoted G_k as:

$$F_k(q, \bar{q}, \hat{q}, P) \rightarrow G_k(\bar{q}, \hat{q}, P) \tag{12}$$

For the actual value of constrained $\hat{q}$ or measured $\bar{q}$ joint variables and kinematic parameters P the Equations (12) have to be zero.

The complexity of the elimination procedure fluctuates according to the available sensor information.

We claim that algebraic methods may be a good alternative to numerical methods. For example, we use in reference [17] a dialytic method to eliminate the orientation variables unmeasured by an external device.

In the case of the constrained variables $\bar{q}$, two types of approach are possible:

- these variables may be eliminated in the constraint equation (for example, by considering the difference between their values in two different configurations),
- or, they are considered as unknown variables whose values will be determined during the identification process.

4.3.2. The *Hexapode 300* calibration cases. For each robot pose, under the hypotheses **H-2**, **H-3** and **H-4**, we eliminate the free joints variables as follows.

(i) M-I case

In method **M-I**, we calibrate the *Hexapode 300* with the help of the pose measurement and leg length measurements. We have to eliminate the free joint variables $q_{i,1}$ and $q_{i,2}$ which model the direction of T_i and $q_{i,4}$, $q_{i,5}$ and $q_{i,6}$ which model the orientation matrix R_{bi} from the implicit system of Equations (9) and (10).

Using Equations (9) and (10), we may eliminate the orientation matrix $R_{ai}\, R_i\, R_{bi}^{-1}$, for $i \in [1, \ldots, 6]$:

$$R_{a0}\, T_0 + R_{a0}\, R_0\, R_{b0}^{-1}\, (b_i - b_0) - (a_i - a_0) = R_{ai}\, T_i \tag{13}$$

The elimination of the matrix R_{ai} is obtained by applying the norm operator on the previous vector; we obtain for each leg $i \in [1, \ldots, 6]$:

$$\| R_{a0}\, T_0 + R_{a0}\, R_0\, R_{b0}^{-1}\, (b_i - b_0) - (a_i - a_0) \|_2 = \| T_i \|_2$$

Then, under Hypothesis **H-1**, the norm of the vector T_i is equal to the prismatic joint variable $q_{i,3} = L_i + l_i$. This allows the elimination of the U-joint variables $q_{i,1}$ and $q_{i,2}$; we obtain for each leg $i \in [1, \ldots, 6]$:

$$\| R_{a0}\, T_0 + R_{a0}\, R_0\, R_{b0}^{-1}\, (b_i - b_0) - (a_i - a_0) \|_2 = (L_i + l_i) \tag{14}$$

We obtain 6 equations per configuration of measurement which are the equations of Inverse Kinematic of Gough platform. To identify the 42 kinematic parameters (see Section 3.5.2), we thus need at least 7 measurement poses ($N \geq 7$). See Figure 2.

The constraint equations are then given as follows:

$$G_{k,i} = \| R_{a0}\, T_{0,k} + R_{a0}\, R_{0,k}\, R_{b0}^{-1}\, (b_i - b_0) - (a_i - a_0) \|_2 - (L_{i,k} + l_i) \tag{15}$$

with the legs' index $i \in [1, \ldots, 6]$ and k the index of the pose robot $k = 1, \ldots, N$ which may be expressed in terms of position $T_{0,k}$ and in terms of orientation $R_{0,k}$ for a command on the leg length $L_{i,k}$.

(ii) M-II case

This *self-calibration* method proposes to instrument all the U-joints of the robot in addition to the leg length. We have to eliminate the 18 free joint variables which model the 6 spherical joints $q_{i,4}$, $q_{i,5}$, $q_{i,6}$ from the initial system (7) of 30 equations.

Using Equation (10) and (9) with $i \neq j \neq 0$ (additionnal leg 0 is not used in the self-calibration case), we get:

$$R_{ai}\, R_i\, R_{bi}^{-1}\, (b_j - b_i) = R_{aj}\, T_j + a_j - R_{ai}\, T_i - a_i \tag{16}$$

The matrix $R_{ai}\, R_i\, R_{bi}^{-1}$ (and then the 18 spherical joint variables) are eliminated as follows.

$$\| R_{ai}\, R_i\, R_{bi}^{-1}\, (b_j - b_i) \|_2 = \| R_{aj}\, T_j + a_j - R_{ai}\, T_i - a_i \|_2$$

$$\| b_j - b_i \|_2^2 = \| R_{aj}\, T_j + a_j - R_{ai}\, T_i - a_i \|_2^2$$

$$\text{with } i, j \in [1, \ldots, 6], \quad i \neq j$$

This system provides 15 equations of which only $30 - 18 = 12$ are independent for each configuration of the robot. To identify the 66 kinematic parameters (see Section 3.5.2), we need at least 6 measurement poses ($N \geq 6$). See Figure 2.

The constraint equations are then given as follows

$$G_{k,i,j} = \| b_j - b_i \|_2^2 - \| R_{aj}\, T_{j,k} + a_j - R_{ai}\, T_{i,k} - a_i \|_2^2 \tag{17}$$

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

with the legs' index $i, j \in [1, \ldots, 6]$, $i \neq j$ and k the index of the pose robot $k = 1, \ldots, N$. For each configuration k, under the hypotheses **H-2**, **H-3**, **H-4** and **H-5**, the vector $T_{i,k}$ (or $T_{j,k}$) is obtained by the translation part of the Equation (1) and the Equations (2), (3), (4) as function of U-joint values $\Theta_{i,k}$, $\Gamma_{i,k}$ and prismatic joint value $L_{i,k}$ which are provided by internal sensors.

(iii) M-III case

This last method uses all the information available to calibrate the robot: the pose measurement and the internal information provided by the sensor installed on U-joints and prismatic joints.

To eliminate the spherical joint variables, we use the same technique as presented previously in the case of M-I. Equation (13) provides directly the constraint equations

$$G_{k,i} = R_{a0}\, T_{0,k} + R_{a0}\, R_{0,k}\, R_{b0}^{-1}\, (b_i - b_0) - (a_i - a_0) - R_{ai}\, T_{i,k} \quad (18)$$

with the legs' index $i \in [1, \ldots, 6]$ and k the index of the pose robot $k = 1, \ldots, N$. For each configuration k, Equations (18) are functions of the robot pose measurements $T_{0,k}$, $R_{0,k}$ and internal sensors $\Theta_{i,k}$, $\Gamma_{i,k}$ and $L_{i,k}$ which provide the value of $T_{i,k}$.

The dimension of $G_{k,i}$ is equal to 3, and then we obtain $3 \times 6 = 18$ constraint equations for one configuration of measurement k. To identify the 78 kinematic parameters (see Section 3.5.2), we need, at least, 5 measurement poses ($N \geq 5$). (Table II).

4.4 Observability of the system

Many kinematic parameters may be identified by a calibration method. But, the only interesting kinematic parameters are those used by the command of the robot. Generally, robot control uses the Inverse Kinematics to put the robot in the wanted pose as a function of the coordinates of the attachment point a_i and b_i and the offset l_i. But it is interesting to identify the others kinematic parameters error to act physically on them during the manufacturing and the assembly of the robot to improve their accuracies.

The problem is to determine what are the relation between the kinematic parameters according to the type of available sensor information. A solution is to process the Identification Jacobian, defined by the derivate of the constraint equations with respect to kinematic parameters P, noted $J_P = (\partial G_k)/(\partial P)$. For a number of constraints greater or equal to the number of parameters considered (7 for **M-I**, 6 for **M-II**, and 5 for **M-III**), the rank of the matrix J_P gives the minimal number of kinematic parameters which may be identified. Besnard and Khalil propose[22] to check the relation between the kinematic parameters through a QR decomposition of this matrix. This method is, in fact, the best solution to check and to correct the robot modeling.

Table II. Minimal number of measurement poses.

Using Method	Kinematic parameters identifiable	No. of equations per pose	Minimal no. of poses
M-I	42	6	7
M-II	66	12	6
M-III	78	18	5

In the case of the *Hexapode 300* calibration, the Identification Jacobian is symbolically computed in Maple. For each method, we note each matrix $J_P^{\text{M-I}}$, $J_P^{\text{M-II}}$ and $J_P^{\text{M-III}}$. Their ranks are checked by considering a number of kinematic parameters equal to 42 for **M-I**, 72 for **M-II** and 84 for **M-III**. We find 42 for method **M-I**, 66 for method **M-II** and 78 for method **M-III**. The result shows that all kinematic parameters are identifiable for method **M-I** but 6 are not observable for the methods **M-II** and **M-III**.

The problem comes from the term "$R_{ai}\, T_{i,k}$" in Equations (17) and (18). First, 3 Euler's angles around the axes **z**, **y** and **x**, define the matrix R_{ai}. We obtain $R_{ai} = \Re_z^{a_i} \circ \Re_y^{a_i} \circ \Re_x^{a_i}$. The vector T_i is obtained with the translation part of the Equation (1) additionally to the Equations (2), (3), (4). We obtain:

$$R_{ai}\, T_i = \Re_z^{a_i} \circ \Re_y^{a_i} \circ \overbrace{\Re_x^{a_i} \circ \Re_x(\theta_i)}^{\text{Parameters are linked}} \circ \Re_x(\Theta_i) \circ \Re_z\left(\frac{\pi}{2} + \alpha_i\right)$$
$$\circ\, \mathcal{T}_z(d_i) \circ \Re_x(\gamma_i) \circ \Re_x(\Gamma_i) \circ \mathcal{T}_z(L_i + l_i) \quad (19)$$

Note that the term $\ldots \Re_x^{a_i} \circ \Re_x(\theta_i) \ldots$ link the parameters describing the last rotation of R_{ai} around the x axis and the offset on $q_{1,i}$ noted θ_i. Those parameters are not identifiable independently. Then, we chose to fix the value of θ_i at an estimation – see Section 5.3. The number of kinematic parameters observable is equal to 66 for method **M-II** and 78 for method **M-III**.

The validity of the robot model is not the only information provided by the matrix J_P. If we use, for example, a least square minimization method (see Section 4.5) to solve a calibration problem, the Identification Jacobian gives the direction of the convergence. It is possible to define several indices associated to this matrix to check the robustness of the parameters identification with respect to measurement noise. In references [23–25, 28], the authors propose to observe the Jacobian J_P through its singular values. 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[25] uses a conjugate-type optimization method. Zhuang[26] tries to avoid local minima by using a Simulated Annealing approach or a Genetic Algorithm.[27] Born and Menq[23] study the determinant of $J_P^T J_P$ to determine how one should modify poses parameters to maximize their index. We propose[29] to adapt an algorithm named DETMAX,[30] widely used in the numerical community for the design of experiments. This algorithm is used to maximize the determinant of $J_P^T J_P$ as a function of the chosen measurement poses. This iterative algorithm, applied on the basic method of Gough platform calibration, converges to a set of poses that are located at the boundary of the joint workspace. A validation through simulation of a calibration experience shows that selected configurations improve significantly the kinematic parameters identification by dividing by 10 to 15 the noise associated to the results.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

4.5. Solving systems of constraint equations

The minimal bound on the number of poses used to calibrate the robot, is given by the dimension of redundant information on the state of the robot (see Section 4.3). The maximal bound is free and only limited by practical considerations. But, experimentally, we can show that there is a limit where the robustness with respect to the noise is not improved.[14]

In any case, the identification of parameters is an "over-constrained" problem: the number of equations is greater than the number of unknowns. This extra information may be used to minimize the influences of the noise associated with sensors on the result of the identification.

The most popular method to solve this kind of problem is to use an algorithm of a non-linear least square method. But, alternative methods are interesting to investigate as algebraic method or interval analysis.[31]

The general description of Least Square algorithm is done as:

$$\text{Minimum}_{P,\hat{q}} \sum_{k=1}^{N} G_k(\bar{q}, \hat{q}, P)^T \, G_k(\bar{q}, \hat{q}, P) \qquad (20)$$

Many methods may be used such as Gradient, Newton, Levenberg-Marquart, Gauss-Newton, etc. All those methods are based on the iterative convergence to a minimum of the constraint equations with respect to unknowns P using a local approximation of those equations by first orders of Taylor series expansion.

The Identification Jacobian matrix J_P, which gives the direction of convergence, is defined as:

$$J_P = \frac{\partial G_k}{\partial P} \qquad (21)$$

Several methods may be applied to our problem. We shall not develop them in full detail. In the case of *Hexapode 300* calibrations, we use a Levenberg-Marquart method implemented by the function *Leastsq* of Matlab 5.2.[32] The system of constraint equations are provided by the Equations (15), (17) or (18). To check the validity of the identification of parameters we simulate by the three methods. A noise is added to the theoretical kinematic parameters P_e to simulate a real set of kinematic parameters P_a. The amplitude of this noise is equal to 5 mm on a_0, b_0, 1 degree on the Euler's angles used in the definition of R_{a0}, R_{b0}, 1 mm on a_i, b_i, l_i, 0.1 degree on R_{ai}, Γ_i, 0.01 mm on d_i and 0.01 degree on α_i. With the help of P_a a simulation of measurement values is constructed. Without adding a noise on measurement simulation values, the calibration methods in the three cases **M-I**, **M-II**, **M-III**, provide exactly the actual kinematic parameters P_a from the estimation P_e.

One may remark that it is possible to fix the reference frame Ω_o and the mobile frame Ω_c to the target Ω_0^a and Ω_0^b and not as reference of the attachment points. That allows to sub-divide the calibration problem for method **M-I** and **M-III** in the sub-calibration of each leg independently. By running a simulation of the two types of method, we noticed that it does not change the identification of the kinematic parameters, but the time of the process of the total method is larger than the time necessary in the sub-divided case. To keep the unified formulation of calibration, we shall present the calibration with the frame definition proposed previously.

The simulations validate our calibration methods. Section 5 presents the results for the experimentations with the real measurement data provided by the sensors.

5. EXPERIMENTATIONS

5.1. Robot description

A copyright on the modeling of the *Hexapode 300* (Figure 7), protects this industrial application. For this reason, we will present the results in terms of relative improvement of the accuracy of the robot.

The theoretical coordinates of attachment points have been calculated so that the wokspace includes a hemisphere in which no singularity can be found.[33]

5.2. Measurement devices

The calibration of the *Hexapode 300* uses 4 types of devices, 3 internal sensors (Figure 8) and one external measurement machinery (Figure 9).

- 6 incremental linear sensors HEIDENHAIN are installed on the 6 prismatic joints to measure the leg length

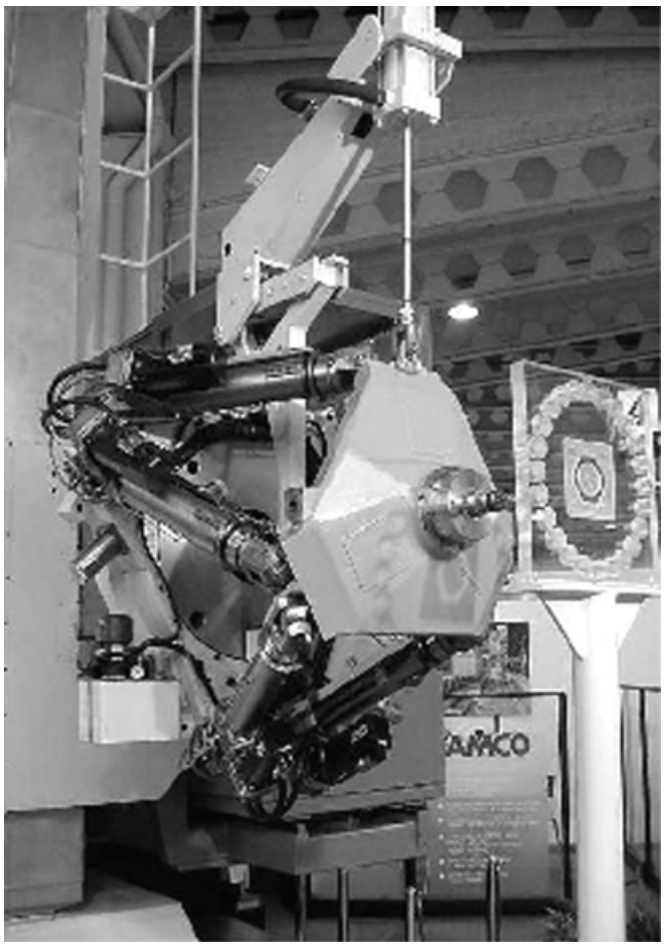


Fig. 7. *Hexapode 300* installed on a support machine.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

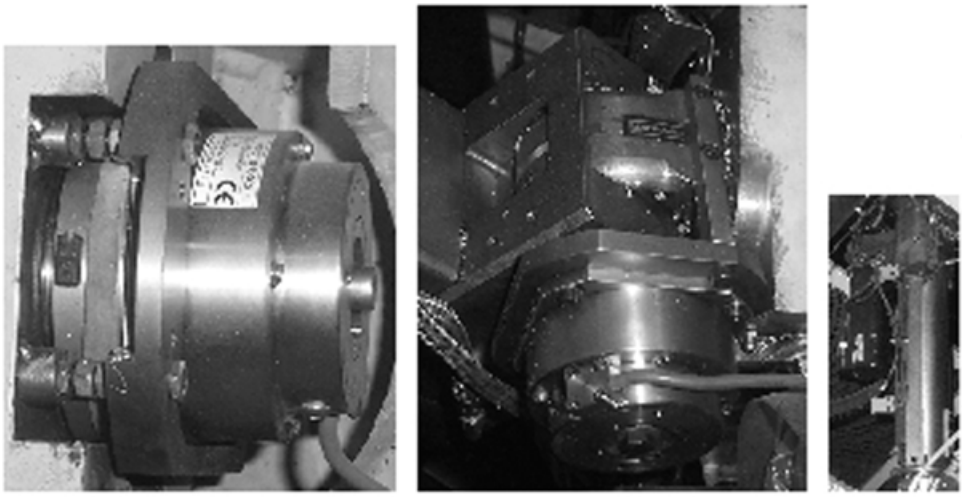


Fig. 8. Measurement devices.

variations. For each sensor, the accuracy given by the constructor is equal to $\pm 4\mu m$. Those measurements are used to control the *Hexapode 300*.

- $2\times3+2\times3$ incremental encoders CODECHAMPS (type I) and RENISHAW RGS-S (type II) installed on 3+3 U-joints. For each encoder, the accuracy is equal to $\pm 0.001°$.
- coordinate measuring machine BROWN&SHARP which provides the 3 coordinates of a point attached to the end-effector of the machine with respect to a reference frame with an accuracy equal to $\pm 5\mu m$.

5.3. Data collection and kinematic parameter estimation

The theoretical values of the kinematic parameters are given by CMW and noted P_0. Some of them are not directly provided, like the offset on the revolute joint θ_i and γ_i.

We put the robot in an initial pose. The sensor outputs Θ_i, Γ_i are fixed to zero in this configuration. Then, we use a Generalized Inverse and a Forward kinematics of the robot to calculate the revolute joint values $q_{i,1}$ and $q_{i,2}$ which are equal to an approximation of the values of θ_i and γ_i.

Now, we want the measurement machinery to provide the position $T_{0,k}$ and the orientation $R_{0,k}$ of the frame Ω_0^a with respect to Ω_0^b for each measurement configurations $k=1,\ldots,N$. For that, those frames Ω_0^a and Ω_0^b must have a physical definition. We construct two accurate targets, see Figure 10, which are composed of a cylinder and two orthogonal planes. For each frame, the direction of the cylinder gives the axis $\mathbf{z}_0^a$ (or $\mathbf{z}_0^b$), the intersection of this axis with the A plane gives the center a_0 (or b_0) and the axis $\mathbf{x}_0^a$ (or $\mathbf{x}_0^b$) is defined such that its direction is parallel to the B plane. Those targets are then installed on the platforms. We use the measurement machinery to calculate an approximation of the localizations of Ω_0^a with respect to Ω_o (in orientation R_{a0} and in position a_0) and Ω_0^b with respect to Ω_c (in orientation R_{b0} and in position b_0) by comparing some reference points on the robot with targets positioning by using the measurement machinery.

The robot is installed inside the workspace of the coordinate measuring machine. We put the robot in various poses. The frames associated with targets Ω_0^a and Ω_0^b are

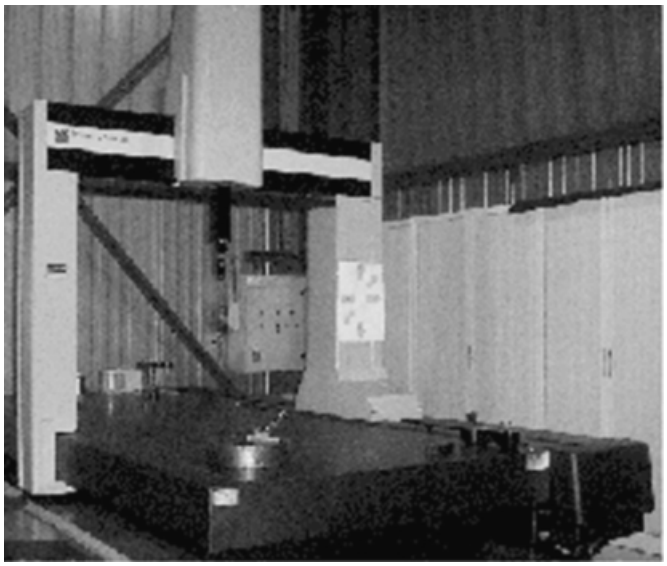


Fig. 9. Coordinate measuring machine.

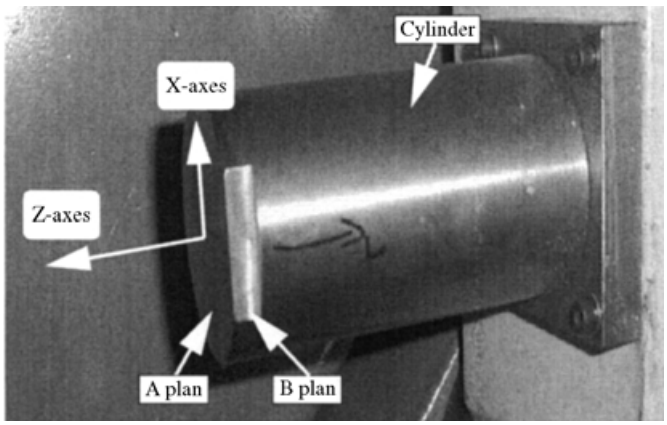
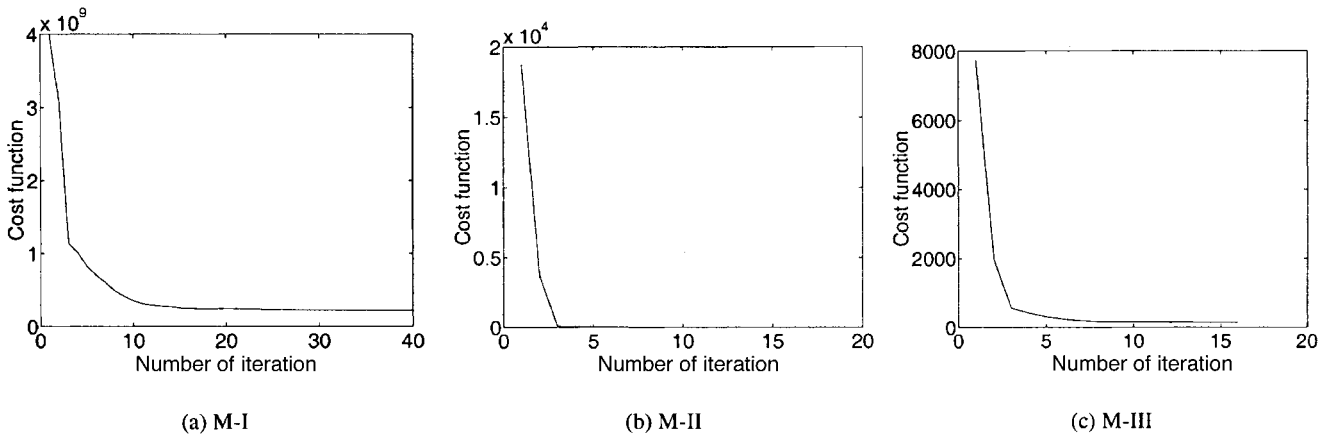


Fig. 10. Two targets are installed on mobile and base platforms.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press



(a) M-I (b) M-II (c) M-III

Fig. 11. Convergence of cost functions of method M-I, M-II, M-III.

reconstructed by measuring the machine software with the help of several points localized on the cylinders and on the planes. From this information we deduce the position $T_{0,k}$ and the orientation $R_{0,k}$ of the frame Ω_0^a with respect to Ω_0^b.

The robot's poses are chosen to optimize the observability of the identification matrix (derivate of the constraint equations with respect of kinematic parameters) by an algorithm of planning experiment given in reference [29]. We put the robot in sixteen poses, $N=16$ (after elimination of out-liners). Additionally, the robot is put in four *validation* poses. The measurement data are not used by the calibration process. This information is only used to validate the method **M-I**, **M-II**, **M-III**.

5.4. Results

With the measurement data provided by sensors for the 16 poses, we process the calibration procedures **M-I**, **M-II**, **M-III** described in Section 4.3.2. The identified kinematic parameters are defined in Section 3.5.2.

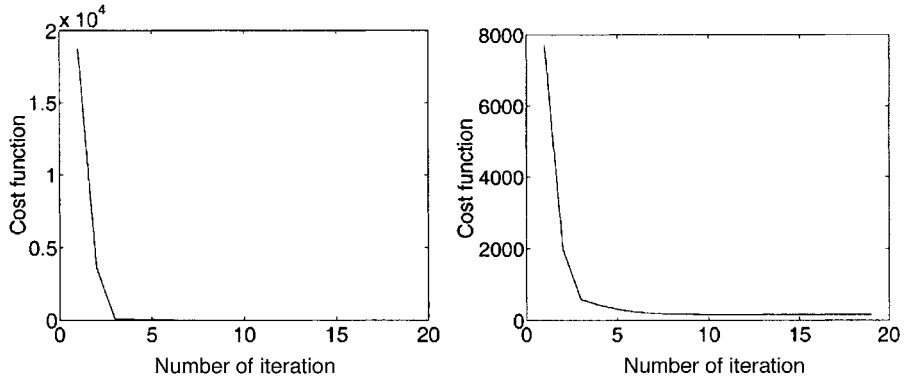
Figure 11 gives the cost function evaluated by least-square methods for methods **M-I**, **M-II** and **M-III** as a function of the number of iterations of this optimization algorithm.

We obtain a good convergence to a minimum for all methods. To check the quality of this minimum, we add a random noise on the theoritical estimation of kinematic parameters $P_0=P_0+\varepsilon$. After a new processing of the method **M-I**, **M-II**, **M-III**, we obtain the same result as in the first experience: $P_c^{\text{M-I}}$, $P_c^{\text{M-II}}$, $P_c^{\text{M-III}}$. We may conclude that those calibration methods are not sensitive to local minima.

The value of each kinematic parameter is compared to a tolerance range given by CMW. Only the kinematic parameters which model the deflect in U-joint, by the method **M-II** and **M-III**, are outside a realistic range (method **M-I** does not allow the identification of those parameters). We conclude that these kinematic parameters cannot be correctly identified. This is due to the noise associated with measurement which is too large to allow the identification of these too small errors. Methods **M-II** and **M-III** are then run again without identifing the 12 parameters d_i and α_i. In this case, Figure 12 gives the cost function evaluation as a function of the number of iterations of **M-II** and **M-III**.

The number of kinematic parameters for each method is now 54 for **M-II** and 66 for **M-III**.

We want to choose the best method to calibrate the robot. The problem is that each method provides a different result: $P_c^{\text{M-I}} \neq P_c^{\text{M-II}} \neq P_c^{\text{M-III}}$. The reason for those differences is that the noise associated with the measurement translates the actual solution to the calibration solution. These results show that it is not possible to conclude on the validity of the



(a) M-II (b) M-III

Fig. 12. Convergence of cost functions of method M-I, M-II, M-III to a minimum in the case where the U-joints deflects are not identified.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

Table III. Cross evaluations of cost functions.

Using Method	P_{initial}	Kinematic parameters $P_c^{\text{M-I}}$	$P_c^{\text{M-II}}$	$P_c^{\text{M-III}}$
M-I	0%	100.00%	97.88%	98.17%
M-II	0%	72.07%	100.00%	99.97%
M-III	0%	54.79%	99.90%	100.00%

calibration methods using only the minimization results; hence others validation methods have to be examined.

5.5 Validations

In this section three solutions are proposed to validate the calibration of the *Hexapode 300*.

The first one consists in evaluating each set of kinematic parameters $P_c^{\text{M-I}}$, $P_c^{\text{M-II}}$, $P_c^{\text{M-III}}$ with the cost function provided with each method **M-I**, **M-II**, **M-III**. The problem is that each method does not identify the same type of kinematic parameters. But, as explained in Section 4.4, the only kinematic parameters which are interesting are those used by the control of the robot: the relative coordinates of attachment points a_i, b_i and the offset on the leg length l_i. We then run three time each method **M-I**, **M-II**, **M-III** by fixing successively the values of a_i, b_i, l_i to those given after calibration $a_i^{\text{M-I}}$, $b_i^{\text{M-I}}$, $l_i^{\text{M-I}}$ or $a_i^{\text{M-II}}$, $b_i^{\text{M-II}}$, $l_i^{\text{M-II}}$ or $a_i^{\text{M-III}}$, $b_i^{\text{M-III}}$, $l_i^{\text{M-III}}$ and by identifying only the other kinematic parameters: R_{a0}, R_{b0}, a_0, b_0 for **M-I** and **M-III**, R_{ai}, γ_i for **M-II** and **M-III**.

We obtain the results given in Table III. It is difficult to compare criteria which do not have the same dimension. To avoid this problem the result is given in terms of improvement of each criterion. In Table III, 0% is the evaluation of the criterion with the kinematic parameters given by CMW P_0; 100% corresponds to the minimal value which may be obtained ($P_c^{\text{M-I}}$ for **M-I**, $P_c^{\text{M-II}}$ for **M-II**, $P_c^{\text{M-III}}$ for **M-III**).

We may note that the set of kinematic parameters provided by the first method, $P_c^{\text{M-I}}$, gives the worst result if we evaluate this set with the criterion of the method **M-II** or **M-III**. In fact, the first method uses less information on the state of the robot compared to the other methods. In terms of dimension, the redundant information is equal to 6 for **M-I** (3 for position and 3 for orientation), 12 for **M-II** (12 revolute joint values) and 18 for **M-III** (3 for position and 3 for orientation + 12 revolute joint values). But the cross evaluation of criterion shows that the results are not clearly better for $P_c^{\text{M-III}}$ than $P_c^{\text{M-II}}$.

Another validation of the method is to check the improvement of the accuracy of the robot with the help of 4 configurations which are not used by the calibration process.

With the help of a numerical Forward Kinematics using a Newton method,[33] we compute the pose error after calibration defined as the norm of the difference between the position and the orientation calculated through the FK for each set of kinematic parameters, $P_c^{\text{M-I}}$, $P_c^{\text{M-II}}$ and $P_c^{\text{M-III}}$ and the position and the orientation provided by the measurement machinery. We compare these values to the same type of error before calibration which is computed with the initial estimation of the kinematic parameters P_0. The results are given in Table IV in terms of improvement of the accuracy before and after calibration. The first four lines present this result for the "validations" poses. The last one presents the mean of the positioning error for the configuration used as input data by the calibration methods. We obtain the same type of results as with the first validation. Method **M-I** gives the worst result and **M-III** the best. But the result given by **M-II** is approximately the same as **M-III** for poses χ_3 and χ_4.

Table IV. Improvement of robot accuracy for validation poses.

Validating poses	Robot modeling	Improvement of accuracy Position [%]	Orientation [%]
χ_1	$P_c^{\text{M-I}}$	90.63	71.95
	$P_c^{\text{M-II}}$	96.92	90.71
	$P_c^{\text{M-III}}$	98.77	91.10
χ_2	$P_c^{\text{M-I}}$	87.95	66.66
	$P_c^{\text{M-II}}$	96.77	90.67
	$P_c^{\text{M-III}}$	98.98	90.48
χ_3	$P_c^{\text{M-I}}$	87.00	71.04
	$P_c^{\text{M-II}}$	96.02	79.73
	$P_c^{\text{M-III}}$	97.70	80.34
χ_4	$P_c^{\text{M-I}}$	92.11	85.04
	$P_c^{\text{M-II}}$	98.58	96.99
	$P_c^{\text{M-III}}$	98.35	97.09
$\chi_{1,\ldots,N}$	$P_c^{\text{M-I}}$	92.50	80.96
	$P_c^{\text{M-II}}$	96.61	81.46
	$P_c^{\text{M-III}}$	97.34	81.79

Using the best set of kinematic parameters, we then test the *Hexapode 300* directly by manufacturing a specific part by the robot. The piece (Figure 13) is a square with an edge equal to 200 mm and is machined in the center of the workspace. The error between the theoretical definition of the object and the machining result is around ±0.01 mm for a tool speed equal to 20 m/minute. Note that not only errors on the kinematic parameters are involved on these errors, as clearly control is involved.

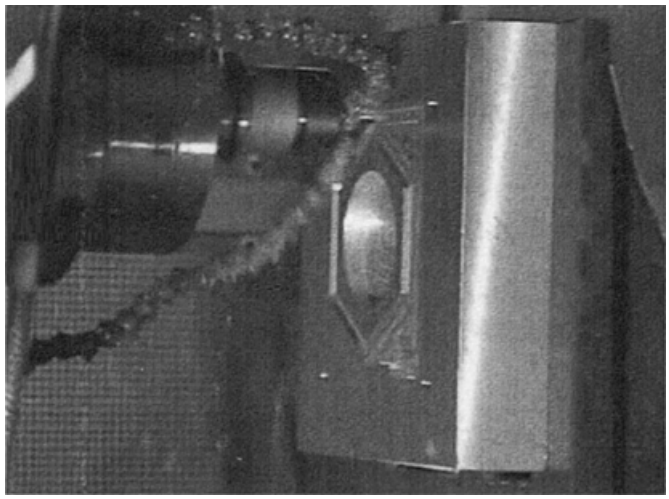


Fig. 13. Reference piece for machining tests.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

5.6. Discussion

The results show that method **M-II** using redundant internal information is better than method **M-I** using the external measurements. An explanation is that the amount of redundant information is larger for **M-II** than for **M-I**. But, we remark that the last method **M-III** does not give a significantly better result than **M-II**. We believe that the noise associated with the pose measurements provided by the external machinery has a large influence on the calibration result. In fact, the position and the orientation of the mobile target frame with respect to the base target frame is not correctly given by the software of the measurement machinery. We think that it is possible to improve the pose measurement by directly using the points which touch the cylinder and the planes and then avoid problems brought by measurement machinery software.

An additional solution is to identify the measurement noise. Wampler[15] proposed this type of method. In our case, to evaluate the noise associated with target reconstruction is difficult. We think that with a simple three balls target, we may have a better control on the measurement values and noise.

In case of an intervention for customer support on the robot, the self-calibration method **M-II** is the solution that will be adopted by CMW. This choice is motivated by:

(i) We have seen that method **M-I** provides less accuracy than method **M-II**.
(ii) The time of experimentation is around half an hour for **M-II** and approximatively 3 days for **M-I** or **M-III**.
(iii) The method **M-II** can be performed automaticaly and it is not necessary to disassemble the robot from the support mechanism.
(iv) The information brought by redundant sensors may be used in the future, to improve the robot accuracy through a simplification of the FK which should be introduced into the robot control.

Validating the accuracy of a machining center with a measurement machinery is not easy from a practical viewpoint. Moreover, the result is as a function of the error on the targets localization parameters R_{a0}, a_0, R_{b0}, b_0 which are not of interest for the control of the robots and thus do not reflect the actual accuracy of the robot.

In the case of a machining center, the validation may be done directly by comparing the theoretical modeling of a part and the manufactured result. But, the error does not provide only the kinematics error but a sum of deflects due to the control, the type of machining, etc. Hence for comparison purpose it will be necessary to define a norm that can be used for parallel structure.

6. CONCLUSION

We have presented an experimentation of the three methods to calibrate a Gough platform. By using the *Implicit Calibration*, we have shown that the basic system of equations may be obtained according only to the available sensor information and independently of the calculation of the kinematics models.

The principle is based on the specification of the closed loop equations of parallel robots. Then, a unified formulation may describe all types of calibration and a strict definition of *self-calibration* and *calibration* is proposed. The perspective of this unified process is to automatically generate the constraint equations only as a function of the description of the information available (measurement or constraints). To eliminate the free joint variables, some algebraic techniques may be used.

To illustrate this formulation, we presented three different methods using external measurement (**M-I**), additional redundant sensors (**M-II**) or both (**M-III**). The first is a basic method, the second is inspired from existing works and the third is completely original. We have seen that the kinematic parameters are well identified as a function of the dimension of the redundant information on the state of the robot. We improve experimentally the accuracy of the robot by 90% for **M-I**, 97% for **M-II**, 99% for **M-III**. We then obtain an error on the manufactured pieces equal to ± 0.001 mm. In the case presented, we show why the self-calibration of the *Hexapode 300* (**M-II**) is the best solution in an industrial context.

Finally, the problem of defining a standard procedure to evaluate both the calibration and the accuracy of a non-linear machining center is emphasized.

Acknowledgements

The author would like to acknowledge Jean-Pierre Merlet (INRIA Sophia) for several helpful discussions, Francois Wildenberg (CMW), Gerard Giraudon (DirDRI INRIA) and Paul Zimmermann (INRIA Lorraine-LORIA) who gave me total liberty to implement this huge theoretical and experimental work and, the Spaces Team (INRIA Lorraine-LORIA) for their various suggestions and their help.

References

1. C. Wampler, and T. Arai, "Calibration of robots having kinematic closed-loops using nonlinear least squares estimator", *IFToMM*, 153–158 (1992).
2. H. Zhuang, "Self-calibration of parallel mechanisms with a case study on Stewart platforms", *IEEE Trans. Robotics Autom.* **13**, 387–397 (1997).
3. L. Baron, "Contributions to the estimation of rigid-body motion under sensor redundancy", *PhD thesis* (CIM, McGill University, 1997).
4. D.J. Bennett, and J.M. Hollerbach, "Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints", *IEEE Trans. Robotics Autom.* **7**, 597–606 (1991).
5. D. Daney, "Self-calibration of Gough platforms using leg mobility constraints", *Proc. 10th World Congress Theory Machines & Mechanisms*, Oulu, Finland (1999) pp. 104–109.
6. W. Khalil, and S. Besnard, "Self calibration of Stewart-Gough parallel robots without extra sensors", *IEEE Trans. Robotics Autom.* **15**, 1116–1121 (1999).
7. H. Zhuang, and Z.S. Roth, "Methods for kinematic calibration of Stewart platforms", *J. Robotic Systems* **10**, 391–405 (1993).
8. Z. Geng, and L.S. Haynes, "An effective kinematics calibration method for Stewart platforms", *ISRAM Proc.* (1994) pp. 87–92.
9. C. Innocenti, "Algorithms for kinematic calibration of fully-parallel manipulators", In: (B. Ravani, J-P. Merlet, eds.), *Computational Kinematics* (Kluwer, 1995) pp. 241–250.

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press

10. O. Masory, H. Zhuang, and Y. Jiahua, "Kinematic calibration of a Stewart platform using pose measurements obtained by a single theodolite", *Proc. IEEE Intell. Robots & Systems* (1995) pp. 329–334.
11. S. Besnard, and W. Khalil, "Calibration of parallel robots using two inclinometers", *Proc. IEEE ICRA*, Detroit, USA (1999) pp. 1758–1763.
12. O. Masory, J. Wang and H. Zhuang, "Kinematic modeling and calibration of a Stewart platform", *Advanced Robotics* **11**, 519–539 (1997).
13. M.P. Oliviers, and J.R.R. Mayer, "Global kinematic calibration of a Stewart platform", *ASME DSC* **57**, 129–136 (1995).
14. H. Zhuang, Y. Jiahua, and O. Masory, "Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals", *J. Robotic Systems* **15**, 395–405 (1998).
15. C.W. Wampler, J.M. Hollerbach, and T. Arai, "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms", *IEEE Trans. on Robotics and Automation* **11**(5), 710–724 (1995).
16. P. Vischer, "Improve the accuracy of parallel robots", *PhD thesis* (Institut des Systèmes Robotiques, École Polytechnique Fédérale de Lausanne, 1996).
17. D. Daney, and I.Z. Emiris, "Robust parallel robot calibration with partial information", *Proc. IEEE ICRA*, Seoul, S. Korea (2001) pp. 3262–3267.
18. J-P. Merlet, M-W. Perng, and D. Daney, "Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator", *Proc. ARK*, Piran (2000) pp. 315–322.
19. I-M. Chen, G. Yang, W.K. Lim, and S.H. Yeo, "Self-Calibration of Three-Legged Modular Reconfigurable Parallel Robots Based on Measurement Residues", *Proc. 2nd Intern. Workshop on Computational Kinematics*, Seoul, S. Korea (2001) pp. 117–132.
20. A. Nahvi, J.M. Hollerbach and V. Hayward, "Calibration of a parallel robot using multiple kinematics closed loops", *Proc. IEEE ICRA*, San Diego (1994) pp. 407–412.
21. J. Denavit and R.S. Hartenberg, "A kinematic notion for lower-pair mechanisms based on matrices", *Transactions of ASME J. of Applied Mechanics* **22**(2), 215–221 (June, 1995).
22. S. Besnard, and W. Khalil, "Identifiable parameters for parallel robots kinematic calibration", *Proc. IEEE ICRA*, Seoul, S. Korea (2001) pp. 2859–2866.
23. J.H. Borm and C.H. Menq, "Determination of optimal measurement configurations for robot calibration based on observabililty measure", *J. of Robotic Systems* **10**(1), 51–63 (1991).
24. M.R. Driels and U.S. Pathre, "Significance of observation strategy on the design of robot calibration experiments", *J. of Robotic Systems* **7**(2), 197–223 (1990).
25. W. Khalil, M. Gautier and Ch. Enguehard, "Identifiable parameters and optimum configurations for robots calibration", *Robotica* **9**, Part 1, 63–70 (1991).
26. H. Zhuang, K. Wang, and Z.S. Roth, "Optimal selection of measurement configurations for robot calibration using simulated annealing", *Proc. IEEE ICRA* (1994) pp. 393–398.
27. H. Zhuang, J. Wu and W. Huang, "Optimal Planning of Robot Calibration Experiments by Genetic Algorithms", *Proc. IEEE ICRA*, Minneapolis (1996) pp. 981–986.
28. A. Nahvi and J.M. Hollerbach, "The noise amplification index for optimal pose selection in robot calibration", *Proc. IEEE ICRA*, Minneapolis (1996) pp. 647–654.
29. T.J. Mitchell, "An algorithm for the construction of *D-optimal* exerimental designs", *Techometrics* **16**, 203–210 (1974).
30. D. Daney, "Optimal Measurement Configurations for Gough Platform Calibration", *Proc. IEEE ICRA*, Washington (11–15 May, 2002) pp. 147–152.
31. J-P, Merlet, "ALIAS: An interval analysis-based library for solving and analyzing systems of equations", *http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS.html*
32. T. Coleman, M. A. Branch, and A. Grace, *Optimization toolbox for use with MATLAB: User's guide* (The Math-Works Inc., 1999, Hawaii, pp. 133–140, 1997).
33. J-P. Merlet, *Parallel Robots* (Kluwer, Dordrecht, 2000).

https://doi.org/10.1017/S0263574703005083 Published online by Cambridge University Press