Chapter 11

Geometric calibration of robots

11.1. Introduction

A high level of positioning accuracy is an essential requirement in a wide range of applications involving industrial robots. This accuracy is affected by geometric factors, such as geometric parameter errors, as well as non-geometric factors, such as flexibility of links and gear trains, gear backlashes, encoder resolution errors, wear, and thermal effects. Positioning accuracy of an industrial robot can be improved to approach its repeatability by a calibration procedure that determines current values of the geometrical dimensions and mechanical characteristics of the structure. Practical techniques to compensate for all geometric and non-geometric effects are not yet developed. Based on investigation of the error contribution from various sources, Judd and Knasinski concluded that the error due to geometric factors accounted for 95% of the total error [Judd 90]. Hence, a reasonable approach would be to calibrate the current geometric parameters and treat the non-geometric factors as a randomly-distributed error. This calibration procedure is also important for robot programming using CAD systems where the simulated robot must reflect accurately the real robot [Craig 93], [Dombre 94], [Chedmail 98]. In recent years, considerable attention has been paid to the problem of geometric calibration. A partial list of these works is given in references [Schefer 82], [Wu 84], [Khalil 85b], [Payannet 85], [Sugimoto 85], [Aldon 86], [Veitschegger 86], [Whitney 86], [Roth 87], [Hollerbach 89], [Mooring 91], [Lavallée 92], [Caenen 93], [Guyot 95], [Damak 96], [Maurine 96], [Besnard 00a].

The problem of geometric calibration can be divided into four distinct steps. The first step is concerned with a particular mathematical formulation that results in a model, which is a function of the geometric parameters ξ, the joint variables $\mathbf{q}$, and eventually some external measurements $\mathbf{x}$. The second step is devoted to the collection of experimental data for a sufficient number of configurations. The third step is concerned with the identification of the geometric parameters and validation of the results obtained. The last step is concerned with compensating the geometric parameter errors in the direct and inverse geometric models.

This chapter addresses all the four steps outlined above for serial robots. The case of the parallel robot is covered briefly at the end.

11.2. Geometric parameters

The geometric parameters concerned with geometric calibration are the parameters required to compute the direct and inverse geometric models. The direct geometric model, which gives the location of the end-effector frame R_{n+1} relative to a fixed world frame R_{-1}, is given by equation [3.13] and is rewritten as:

$$^{-1}\mathbf{T}_{n+1} = \mathbf{Z}\,^{0}\mathbf{T}_n(\mathbf{q})\,\mathbf{E} = {}^{-1}\mathbf{T}_0\,^{0}\mathbf{T}_1\,^{1}\mathbf{T}_2 \ldots {}^{n-1}\mathbf{T}_n\,^{n}\mathbf{T}_{n+1} \qquad [11.1]$$

where:

- $\mathbf{Z} = {}^{-1}\mathbf{T}_0$ denotes the transformation matrix defining the robot base frame R_0 relative to the world reference frame R_{-1};
- $\mathbf{E} = {}^{n}\mathbf{T}_{n+1}$ is the transformation matrix defining the end-effector frame with respect to the terminal link frame R_n;
- $^{0}\mathbf{T}_n$ is the transformation matrix of the robot defining frame R_n relative to frame R_0.

Equation [11.1] contains three kinds of parameters: the robot geometric parameters appearing in $^{0}\mathbf{T}_n$, the base frame parameters defining the matrix $\mathbf{Z}$ and the end-effector parameters defining the matrix $\mathbf{E}$. We add to these parameters the joint gear transmission ratios that can be calibrated in the same manner as the geometric parameters.

For convenience, in the remainder of the chapter, we denote the origin O_{n+1} of the end-effector frame R_{n+1} as the *endpoint* of the robot.

11.2.1. *Robot parameters*

The robot parameters are deduced from the notations developed in Chapter 3. According to these notations, frame R_j is fixed with link j. It is located relative to frame R_{j-1} by the homogeneous transformation matrix $^{j-1}\mathbf{T}_j$, which is a function of the four geometric parameters (α_j, d_j, θ_j, r_j) (Figure 3.2), such that:

$$^{j-1}\mathbf{T}_j = \mathbf{Rot}(\mathbf{x}, \alpha_j)\,\mathbf{Trans}(\mathbf{x}, d_j)\,\mathbf{Rot}(\mathbf{z}, \theta_j)\,\mathbf{Trans}(\mathbf{z}, r_j) \qquad [11.2]$$

Note that the parameters α_1 and d_1 can be taken to be equal to zero by assigning the base frame R_0 aligned with frame R_1 when $q_1 = 0$.

If two consecutive joint axes $j-1$ and j are parallel, the $\mathbf{x}_{j-1}$ axis is taken arbitrarily along one of the common normals between them. When $\mathbf{z}_{j-1}$ or $\mathbf{z}_j$ becomes slightly misaligned, the common normal is uniquely defined and the corresponding variation in the parameter r_{j-1} can be very large. To ensure that small variation in axis alignment produces proportionally small variations in the parameters, we make use of a fifth parameter β_j [Hayati 83] representing a rotation around the $\mathbf{y}_{j-1}$ axis. The general transformation matrix ${}^{j-1}\mathbf{T}_j$ becomes:

$$
{}^{j-1}\mathbf{T}_j = \mathbf{Rot}(\mathbf{y}, \beta_j)\,\mathbf{Rot}(\mathbf{x}, \alpha_j)\,\mathbf{Trans}(\mathbf{x}, d_j)\,\mathbf{Rot}(\mathbf{z}, \theta_j)\,\mathbf{Trans}(\mathbf{z}, r_j) \qquad [11.3]
$$

The nominal value of β_j is zero. If $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ are not parallel, β_j is not identifiable. We note that when $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ are parallel, we can identify either r_{j-1} or r_j (§ 11.4.2), thus the number of identifiable parameters for each frame is at most four.

11.2.2. *Parameters of the base frame*

Since the reference frame can be chosen arbitrarily by the user, six parameters are needed to locate the robot base relative to the world frame. As developed in § 7.2, these parameters can be taken as $(\gamma_z, b_z, \alpha_z, d_z, \theta_z, r_z)$ (Figure 11.1):

$$
\mathbf{Z} = {}^{-1}\mathbf{T}_0 = \mathbf{Rot}(\mathbf{z}, \gamma_z)\,\mathbf{Trans}(\mathbf{z}, b_z)\,\mathbf{Rot}(\mathbf{x}, \alpha_z)\,\mathbf{Trans}(\mathbf{x}, d_z)\,\mathbf{Rot}(\mathbf{z}, \theta_z)\,\mathbf{Trans}(\mathbf{z}, r_z) \qquad [11.4]
$$

The transformation matrix ${}^{-1}\mathbf{T}_1$ is given by:

$$
{}^{-1}\mathbf{T}_1 = {}^{-1}\mathbf{T}_0\,{}^{0}\mathbf{T}_1 = \mathbf{Rot}(\mathbf{z}, \gamma_z)\,\mathbf{Trans}(\mathbf{z}, b_z)\,\mathbf{Rot}(\mathbf{x}, \alpha_z)\,\mathbf{Trans}(\mathbf{x}, d_z)\,\mathbf{Rot}(\mathbf{z}, \theta_z)\,\mathbf{Trans}(\mathbf{z}, r_z)\,\mathbf{Rot}(\mathbf{x}, \alpha_1)\,\mathbf{Trans}(\mathbf{x}, d_1)\,\mathbf{Rot}(\mathbf{z}, \theta_1)\,\mathbf{Trans}(\mathbf{z}, r_1) \qquad [11.5]
$$

Since $\alpha_1 = 0$ and $d_1 = 0$, we can write that:

$$
{}^{-1}\mathbf{T}_0\,{}^{0}\mathbf{T}_1 = \mathbf{Rot}(\mathbf{x}, \alpha_0)\,\mathbf{Trans}(\mathbf{x}, d_0)\,\mathbf{Rot}(\mathbf{z}, \theta_0)\,\mathbf{Trans}(\mathbf{z}, r_0)\,\mathbf{Rot}(\mathbf{x}, \alpha'_1)\,\mathbf{Trans}(\mathbf{x}, d'_1)\,\mathbf{Rot}(\mathbf{z}, \theta'_1)\,\mathbf{Trans}(\mathbf{z}, r'_1) \qquad [11.6]
$$

with $\alpha_0 = 0$, $d_0 = 0$, $\theta_0 = \gamma_z$, $r_0 = b_z$, $\alpha'_1 = \alpha_z$, $d'_1 = d_z$, $\theta'_1 = \theta_1 + \theta_z$, $r'_1 = r_1 + r_z$

Equation [11.6] represents two transformations having the same kind of parameters as those of equation [11.2]. We note that the parameters θ_z and r_z are grouped with θ_1 and r_1 respectively. This proves that consecutive frames are represented at most by four independent parameters.

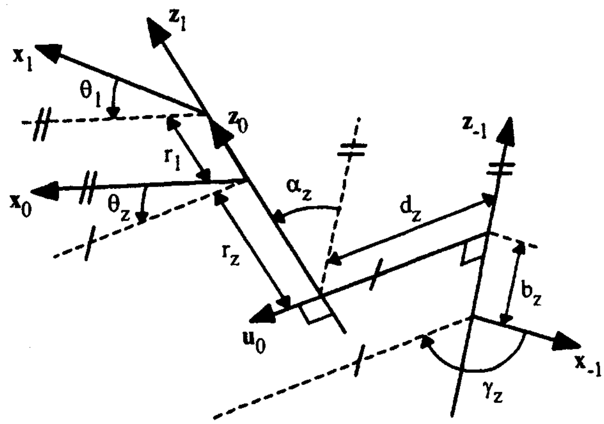

Figure 11.1. *Description of frame R_0 relative to frame R_{-1}*

11.2.3. *End-effector parameters*

Since the end-effector frame R_{n+1} can be defined arbitrarily with respect to the terminal link frame R_n, six parameters (γ_e, b_e, α_e, d_e, θ_e, r_e) are needed to define the matrix **E**. As previously, we can extend the robot notations to the definition of the end-effector frame:

$$^{n}\mathbf{T}_{n+1} = \mathbf{Rot}(\mathbf{z}, \gamma_e)\ \mathbf{Trans}(\mathbf{z}, b_e)\ \mathbf{Rot}(\mathbf{x}, \alpha_e)\ \mathbf{Trans}(\mathbf{x}, d_e)\ \mathbf{Rot}(\mathbf{z}, \theta_e)\ \mathbf{Trans}(\mathbf{z}, r_e) \qquad [11.7]$$

The transformation matrix $^{n-1}\mathbf{T}_{n+1}$ can be written as:

$$\begin{aligned} ^{n-1}\mathbf{T}_{n+1} &= {}^{n-1}\mathbf{T}_n\, {}^{n}\mathbf{T}_{n+1} \\ &= \mathbf{Rot}(\mathbf{x}, \alpha_n)\ \mathbf{Trans}(\mathbf{x}, d_n)\ \mathbf{Rot}(\mathbf{z}, \theta_n)\ \mathbf{Trans}(\mathbf{z}, r_n)\ \mathbf{Rot}(\mathbf{z}, \gamma_e) \\ &\quad \mathbf{Trans}(\mathbf{z}, b_e)\ \mathbf{Rot}(\mathbf{x}, \alpha_e)\ \mathbf{Trans}(\mathbf{x}, d_e)\ \mathbf{Rot}(\mathbf{z}, \theta_e)\ \mathbf{Trans}(\mathbf{z}, r_e) \end{aligned} \qquad [11.8]$$

which gives:

$$^{n-1}\mathbf{T}_{n+1} = \mathbf{Rot}(\mathbf{x}, \alpha_n)\ \mathbf{Trans}(\mathbf{x}, d_n)\ \mathbf{Rot}(\mathbf{z}, \theta'_n)\ \mathbf{Trans}(\mathbf{z}, r'_n)\ \mathbf{Rot}(\mathbf{x}, \alpha_{n+1})\ \mathbf{Trans}(\mathbf{x}, d_{n+1})\ \mathbf{Rot}(\mathbf{z}, \theta_{n+1})\ \mathbf{Trans}(\mathbf{z}, r_{n+1}) \qquad [11.9]$$

with $\theta'_n = \theta_n + \gamma_e$, $r'_n = r_n + b_e$, $\alpha_{n+1} = \alpha_e$, $d_{n+1} = d_e$, $\theta_{n+1} = \theta_e$, $r_{n+1} = r_e$.

Thus, the end-effector frame introduces four independent parameters α_e, d_e, θ_e, r_e, whereas the parameters γ_e and b_e are grouped with θ_n and r_n respectively.

Finally, the description of the location of the end-effector frame R_{n+1} in the reference frame R_{-1} of an n degree-of-freedom robot, needs at most $(4n + 6)$ independent parameters. More precisely, since in the case of a prismatic joint only two parameters can be identified, the maximum number of independent parameters reduces to $(4n_r + 2n_p + 6)$, where n_r and n_p are the numbers of revolute and prismatic joints of the robot respectively [Everett 88].

11.3. Generalized differential model of a robot

The generalized differential model provides the differential variation of the location of the end-effector as a function of the differential variation of the geometric parameters. It is represented by:

$$\Delta X = \begin{bmatrix} dP_{n+1} \\ \delta_{n+1} \end{bmatrix} = \Psi \, \Delta\xi \tag{11.10}$$

with:

- dP_{n+1}: (3x1) differential translation vector of the origin O_{n+1};
- δ_{n+1}: (3x1) differential rotation vector of frame R_{n+1};
- Ψ: $(6 x N_{par})$ generalized Jacobian matrix;
- $\Delta\xi$: $(N_{par} x 1)$ vector of the differential variation of the geometric parameters.

The columns of the generalized Jacobian matrix Ψ can be computed using simple vector relationships as we did in Chapter 5 for the computation of the base Jacobian matrix. According to the kind of parameter, we obtain [Khalil 89c]:

i) column corresponding to the parameter $\Delta\beta_i$: the parameter β_i represents a rotation about the y_{i-1} axis. A differential variation on β_i generates a differential position on frame R_{n+1} equal to $(\mathbf{n}_{i-1} \times \mathbf{L}_{i-1,n+1}) \, \Delta\beta_i$ and a differential orientation equal to $\mathbf{n}_{i-1} \, \Delta\beta_i$. The column of Ψ corresponding to the parameter $\Delta\beta_i$ is given by:

$$\Psi\beta_i = \begin{bmatrix} \mathbf{n}_{i-1} \times \mathbf{L}_{i-1,n+1} \\ \mathbf{n}_{i-1} \end{bmatrix} \tag{11.11}$$

where $\mathbf{L}_{i-1,n+1}$ is the vector connecting O_{i-1} to O_{n+1}, and $\mathbf{n}_{i-1}$ is the unit vector along the y_{i-1} axis;

ii) column corresponding to the parameter $\Delta\alpha_i$: the parameter α_i represents a rotation about the $\mathbf{x}_{i-1}$ axis. A differential variation on α_i generates a differential position on frame R_{n+1} equal to $(\mathbf{s}_{i-1} \times \mathbf{L}_{i-1,n+1})\,\Delta\alpha_i$ and a differential orientation equal to $\mathbf{s}_{i-1}\,\Delta\alpha_i$. The column of $\boldsymbol{\Psi}$ corresponding to the parameter $\Delta\alpha_i$ is given by:

$$\boldsymbol{\Psi}\alpha_i = \begin{bmatrix} \mathbf{s}_{i-1} \times \mathbf{L}_{i-1,n+1} \\ \mathbf{s}_{i-1} \end{bmatrix} \tag{11.12}$$

where $\mathbf{s}_{i-1}$ is the unit vector along the $\mathbf{x}_{i-1}$ axis;

iii) column corresponding to the parameter Δd_i: the parameter d_i represents a translation along the $\mathbf{x}_{i-1}$ axis. A differential variation on d_i generates a differential position on frame R_{n+1} equal to $\mathbf{s}_{i-1}\,\Delta d_i$, but produces no differential orientation. Thus, the corresponding column is expressed as:

$$\boldsymbol{\Psi}d_i = \begin{bmatrix} \mathbf{s}_{i-1} \\ \mathbf{0}_{3x1} \end{bmatrix} \tag{11.13}$$

iv) column corresponding to the parameter $\Delta\theta_i$: this case has been handled in Chapter 5 while calculating the base Jacobian matrix. The corresponding column is given by:

$$\boldsymbol{\Psi}\theta_i = \begin{bmatrix} \mathbf{a}_i \times \mathbf{L}_{i,n+1} \\ \mathbf{a}_i \end{bmatrix} \tag{11.14}$$

where $\mathbf{a}_i$ is the unit vector along the $\mathbf{z}_i$ axis;

v) column corresponding to the parameter Δr_i: this case has also been developed in Chapter 5. The corresponding column is given by:

$$\boldsymbol{\Psi}r_i = \begin{bmatrix} \mathbf{a}_i \\ \mathbf{0}_{3x1} \end{bmatrix} \tag{11.15}$$

vi) column corresponding to a differential variation in the gear transmission ratio: Let us denote $K_i = 1/N_i$, where N_i is the gear transmission ratio. In general, the joint variables are given by:

$$\mathbf{q} = \mathbf{C}_a\,\mathbf{diag}(K_1, \ldots, K_n)\,\mathbf{C}_m\,\mathbf{q}_m + \mathbf{q}_0 \tag{11.16}$$

with:

- $\mathbf{C}_m$: (nxn) matrix representing the coupling between the motor variables;

- $\mathbf{C_a}$: (nxn) matrix representing the coupling between the variables after the gear transmission;
- $\mathbf{q_m}$: vector of motor variables;
- $\mathbf{q_0}$: constant vector representing the offset values on the joint variables.

Thus, the column corresponding to K_i is given by:

$$\Psi_{ki} = \mathbf{J}_{n+1} \frac{\partial \mathbf{q}}{\partial K_i} \qquad [11.17]$$

where $\mathbf{J}_{n+1}$ is the base Jacobian matrix of the robot (§ 5.3) whose i^{th} column is either Ψr_i if joint i is prismatic or $\Psi\theta_i$ if joint i is revolute.

If the joints are actuated independently, we have:

$$\begin{cases} \Psi k_i = q_{mi}\,\Psi r_i & \text{if joint i is prismatic} \\ \Psi k_i = q_{mi}\,\Psi\theta_i & \text{if joint i is revolute} \end{cases} \qquad [11.18]$$

NOTE.– All the vectors used in the computation of Ψ are derived from the matrices $^{-1}\mathbf{T}_i$, $i = 0, \ldots, n+1$. The vectors $^{-1}\mathbf{s}_i$, $^{-1}\mathbf{n}_i$ and $^{-1}\mathbf{a}_i$ are obtained directly from these matrices, whereas the vector $\mathbf{L}_{i,n+1}$ is computed using the following equation:

$$^{-1}\mathbf{L}_{i,n+1} = {}^{-1}\mathbf{P}_{n+1} - {}^{-1}\mathbf{P}_i \qquad [11.19]$$

11.4. Principle of geometric calibration

The calibration of the geometric parameters is based on estimating the parameters minimizing the difference between a function of the real robot variables and its mathematical model. The methods proposed in the literature differ according to the variables used to define this function. In § 11.5, we will present some of these methods. In the following section, they are formulated using a unified approach.

11.4.1. *General calibration model*

The calibration model can be represented by the general nonlinear equation [11.20] and by the general linearized equation [11.21]:

$$\mathbf{0} = \mathbf{f}(\mathbf{q}, \mathbf{x}, \xi) \qquad [11.20]$$

$$\Delta\mathbf{y}(\mathbf{q}, \mathbf{x}, \xi) = \phi(\mathbf{q}, \xi)\, \Delta\xi \qquad [11.21]$$

where:

- $\mathbf{x}$ represents the external measured variables, such as the Cartesian variables giving the position and orientation of the end-effector frame, or the distance traveled by the endpoint between two configurations;
- $\mathbf{q}$ is the (nx1) vector of the joint variables;
- ξ is the (N_{par}x1) vector of the geometric parameters;
- ϕ is the (pxN$_{par}$) calibration Jacobian matrix, whose elements are computed as functions of the generalized Jacobian matrix Ψ;
- $\Delta\mathbf{y}$ is the (px1) prediction error vector.

To estimate $\Delta\xi$, we apply equation [11.20] and/or equation [11.21] for a sufficient number of configurations. Combining all the equations results in the following nonlinear and linear systems of (pxe) equations, where e is the number of configurations:

$$\mathbf{0} = \mathbf{F}(\mathbf{Q}_t, \mathbf{X}_t, \xi) + \rho' \qquad [11.22]$$

$$\Delta\mathbf{Y} = \mathbf{W}(\mathbf{Q}_t, \xi)\, \Delta\xi + \rho \qquad [11.23]$$

with:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}(\mathbf{q}^1, \mathbf{x}^1, \xi) \\ \cdots \\ \mathbf{f}(\mathbf{q}^e, \mathbf{x}^e, \xi) \end{bmatrix} \qquad [11.24]$$

$$\Delta\mathbf{Y} = \begin{bmatrix} \Delta\mathbf{y}^1(\mathbf{q}^1, \mathbf{x}^1, \xi) \\ \cdots \\ \Delta\mathbf{y}^e(\mathbf{q}^e, \mathbf{x}^e, \xi) \end{bmatrix} \qquad [11.25]$$

where $\mathbf{Q}_t = [\mathbf{q}^{1T} \ldots \mathbf{q}^{eT}]^T$, $\mathbf{X}_t = [\mathbf{x}^{1T} \ldots \mathbf{x}^{eT}]^T$ and $\mathbf{W}$ is the (rxN_{par}) observation matrix. ρ and ρ' are the modeling error vector for the nonlinear and linear models respectively, including the effects of unmodeled non-geometric parameters:

$$\mathbf{W} = \begin{bmatrix} \phi^1(q^1, \xi) \\ \cdots \\ \phi^e(q^e, \xi) \end{bmatrix} \qquad [11.26]$$

The number of configurations e must be chosen such that the number of equations, r = pxe, is greater than N_{par}. In practice, good results can be obtained by taking $r \geq 5N_{par}$ and by choosing configurations optimizing the observability measure (§ 11.4.2.2).

Equation [11.22] and/or equation [11.23] can be used to estimate the geometric parameters. However, before solving these equations, we have to rewrite them such that the unknown vector is only composed of the identifiable parameters. Of course, if a parameter is exactly known, it will not be included in the unknown vector ξ.

NOTES.–

- the errors corresponding to the joint variables represent the joint offset errors;
- the number of equations of the function **f** is also called the *calibration index* [Hollerbach 96];
- in autonomous calibration methods, both **f** and $\Delta\mathbf{y}$ are computed in terms of the joint variables and the robot parameters. No external measuring device is needed.

11.4.2. *Identifiability of the geometric parameters*

It may happen that some parameters are not uniquely determined by the identification equation. All sources of parameter ambiguity can be linked to the rank of the matrix **W**. If some columns of **W** are linearly dependent, then the corresponding parameters may vary arbitrarily such that these variations only satisfy the linear dependence. For example, in the conventional calibration method, which uses the measurements of the end-effector location or position (§ 11.5.1), if the joint axes i – 1 and i are parallel, then the columns corresponding to the parameters r_{i-1} and r_i are equal in the calibration Jacobian matrix ($\Psi r_{i-1} = \Psi r_i$). Thus, an infinite set of solutions can be obtained for the errors in r_{i-1} and r_i. The basic solution consists of identifying the error in one of these parameters while assuming that the error in the other parameter is zero.

The loss of identifiability of some parameters may be caused by two kinds of problems, namely structural identifiability and selection of calibration configurations. Unidentifiability of some parameters constitutes a structural problem when some columns of the observation matrix are zero or are linearly dependent whatever the number and the values of the configurations used to construct the observation matrix. The structurally unidentifiable parameters depend on the calibration model and on the structure of the robot. In § 11.4.2.1, we will give an algorithm to determine the structurally identifiable parameters.

The problem of the identifiability as a function of the calibration configurations is known as the *excitation problem*. It will be addressed in § 11.4.2.2 by selecting the calibration configurations that optimize an observability measure. For example, it is obvious that if one joint has a constant value in all the calibration configurations, some parameters concerning this joint will not be identified.

The determination of the identifiable geometric parameters is based on the determination of the independent columns of the calibration Jacobian matrix ϕ. A symbolic method is presented in [Khalil 91b] to determine these parameters for the conventional calibration methods. However, in the following, we develop a general method based on the QR decomposition to determine the identifiable parameters. This method is similar to that which has been presented in Appendix 5 for the determination of the dynamic base parameters.

11.4.2.1. *Determination of the identifiable parameters*

Numerically, the study of the identifiable parameters, also termed *base geometric parameters*, is equivalent to the study of the space spanned by the columns of an (rxNpar) matrix **W** similar to that defined in equation [11.23] but obtained using random configurations satisfying the constraints of the calibration method. The identifiable parameters are determined through the following three steps:

- if a column of **W** is zero, then the corresponding parameter has no effect on the geometric calibration model. Eliminating such parameters and the corresponding columns reduces **W** to an (rxc) matrix; for convenience, we continue to indicate this new matrix by **W**;
- the rank of **W**, denoted by b, gives the number of identifiable parameters;
- a set of identifiable parameters can be chosen as those corresponding to b independent columns of **W**. The other parameters are not identifiable.

To carry out the last two steps, we make use of the QR decomposition of the (rxc) matrix **W**. The matrix **W** can be written as [Dongarra 79], [Lawson 74], [Golub 83]:

$$\mathbf{W} = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(r-c)xc} \end{bmatrix} \tag{11.27}$$

where **Q** is an (rxr) orthogonal matrix, **R** is a (cxc) upper triangular matrix, and $\mathbf{0}_{ixj}$ is the (ixj) null matrix.

Theoretically, the non-identifiable parameters are those whose corresponding elements on the diagonal of the matrix **R** are zero. In practice, they are defined using a numerical tolerance $\tau \neq 0$ [Forsythe 77]. Thus, if $|R_{ii}|$, which represents the absolute value of the (i, i) element of **R**, is less than τ, the corresponding parameter is not identifiable. The numerical zero τ can be taken as [Dongarra 79]:

$$\tau = r.\varepsilon.\max |R_{ii}| \qquad [11.28]$$

where ε is the computer precision, and r is the number of rows.

It is obvious that the identifiable parameters are not uniquely defined. The QR method will provide as base parameters those corresponding to the first b independent columns of the matrix **W**. It is convenient to identify, if possible, the parameters that can be updated in the control system without changing the closed-form geometric models of the robot. Therefore, we permute the columns of **W** and the elements of ξ to first place the following parameters:

- the joint offsets and the gear transmission ratios;
- the parameters r_j and d_j whose nominal values are not zero;
- the angles α_j and θ_j whose nominal values are not $k\,\pi/2$, where k is an integer;
- the parameters defining the matrices **Z** and **E**.

Having determined the identifiable parameters, the linearized identification equation [11.23] is rewritten as:

$$\Delta\mathbf{Y} = \mathbf{W}_b \Delta\xi_b + \rho \qquad [11.29]$$

$\mathbf{W}_b$ is composed of the columns of **W** corresponding to the identifiable parameters. The nonlinear model will be solved in ξ_b. In the remainder of the chapter, the subscript b will be dropped for simplicity. Hence, **W** and $\Delta\xi$ will stand for $\mathbf{W}_b$ and $\Delta\xi_b$ respectively.

11.4.2.2. *Optimum calibration configurations*

The goal is to select a set of robot configurations that yield maximum observability of the model parameters and minimize the effect of noise on the parameter estimation. The condition number of the observation matrix **W** (Appendix 4) gives a good estimate of the observability of the parameters [Driels 90], [Khalil 91b]. Thus, optimum calibration configurations provide a condition number of **W** close to one. We have either to determine the calibration configurations by solving a nonlinear optimization problem that minimizes the condition number, or to verify that the randomly collected data give a good condition number.

The optimization problem has (exn) unknowns (n is the number of joints, e is the number of configurations). It can be formulated as follows [Khalil 91b]:

> Find the configurations $\mathbf{Q}_t = [\mathbf{q}^{1T} \dots \mathbf{q}^{eT}]^T$
> minimizing the criterion $C(\mathbf{Q}_t) = \text{cond}[\mathbf{W}(\mathbf{Q}_t)] = \|\mathbf{W}\|.\|\mathbf{W}^{+}\|$,
> under the calibration method constraints and the following joint limit constraints:
> $q_{i,min} \leq \mathbf{Q}_t\,[i + (j-1)n] \leq q_{i,max}$ where $i = 1, \dots, n$ and $j = 1, \dots, e$

where $\mathbf{q}^j$ is the (nx1) joint position vector corresponding to configuration j, $\mathbf{W}^{+}$ is the pseudoinverse of $\mathbf{W}$, $\|\mathbf{W}\|$ is a norm of $\mathbf{W}$, and $q_{i,min}$, $q_{i,max}$ give the minimum and maximum values of the position of joint i respectively.

We recall that, when using the 2-norm, the condition number is given by (Appendix 4):

$$\text{cond}_2(\mathbf{W}) = \frac{\sigma_{max}}{\sigma_{min}} \qquad [11.30]$$

where σ_{max} and σ_{min} are the largest and smallest singular values of $\mathbf{W}$.

This algorithm has been applied in [Khalil 91b] for the conventional calibration methods that use the Cartesian end-effector coordinates (§ 11.5.1). The optimization algorithm was based on the gradient conjugate method proposed by Powell [Powell 64].

Other observability measures have been proposed in the literature, namely the smallest singular value [Nahvi 94], and the product of the singular values of $\mathbf{W}$ [Borm 91], but the condition number is shown to be more efficient [Hollerbach 95].

It is worth noting that most of the geometric calibration methods give an acceptable condition number using random configurations [Khalil 00b].

11.4.3. *Solution of the identification equation*

The calibrated geometric parameters are obtained by solving the nonlinear algebraic equation [11.22] in order to minimize the least-squares error:

$$\hat{\xi} = \min_{\Delta\xi} \|\mathbf{F}\|^2$$

This optimization problem can be performed using the Levenberg-Marquardt algorithm, which is implemented in Matlab.

For rigid robot calibration, the linearized model can be used to solve iteratively this nonlinear optimization problem. Equation [11.23] is solved to get the least-squares error solution to the current parameter estimate. This procedure is iterated until the variation $\Delta\xi$ approaches zero and the parameters have converged to some stable value. At each iteration, the geometric parameters are updated by adding $\Delta\hat{\xi}$ to the current value of ξ. The observation matrix $\mathbf{W}$ and the prediction error $\Delta\mathbf{Y}$ are updated as well.

The least-squares solution $\Delta\hat{\xi}$ of equation [11.23] is written as:

$$\Delta\hat{\xi} = \min_{\Delta\xi} \|\rho\|^2 = \min_{\Delta\xi} \|\Delta\mathbf{Y} - \mathbf{W}\,\Delta\hat{\xi}\|^2 \qquad [11.31]$$

The solution can be obtained using the pseudoinverse matrix (Appendix 4):

$$\Delta\hat{\xi} = \mathbf{W}^{+}\,\Delta\mathbf{Y} \qquad [11.32]$$

where $\mathbf{W}^{+}$ denotes the pseudoinverse matrix of $\mathbf{W}$. If $\mathbf{W}$ is of full rank, the explicit computation of $\mathbf{W}^{+}$ is given by $(\mathbf{W}^T\,\mathbf{W})^{-1}\,\mathbf{W}^T$.

In general, for rigid robots, the iterative least-squares method converges much faster than the Levenberg-Marquardt algorithm.

Standard deviation of the parameter estimation errors is calculated using the matrix $\mathbf{W}$ as a function of the estimated geometric parameters. Assuming that $\mathbf{W}$ is deterministic, and ρ is a zero mean additive independent noise with standard deviation σ_ρ, the variance-covariance matrix $\mathbf{C}_\rho$ is given by:

$$\mathbf{C}_\rho = E(\rho\,\rho^T) = \sigma_\rho^2\,\mathbf{I}_r \qquad [11.33]$$

where E is the expectation operator and $\mathbf{I}_r$ is the (rxr) identity matrix.

An unbiased estimation of σ_ρ can be computed using the following equation:

$$\sigma_\rho^2 = \frac{\|\Delta\mathbf{Y} - \mathbf{W}\,\Delta\hat{\xi}\|^2}{(r - c)} \qquad [11.34]$$

The variance-covariance matrix of the estimation error is given by [de Larminat 77]:

$$\mathbf{C}_{\hat{\xi}} = E[(\xi - \hat{\xi})\,(\xi - \hat{\xi})^T] = \mathbf{W}^{+}\,\mathbf{C}_\rho\,(\mathbf{W}^{+})^T = \sigma_\rho^2\,(\mathbf{W}^T\,\mathbf{W})^{-1} \qquad [11.35]$$

The standard deviation of the estimation error on the j^{th} parameter is obtained from the (j, j) element of $\mathbf{C}_{\xi}$:

$$\sigma_{\xi_j} = \sqrt{C_{\xi}(j,j)} \qquad [11.36]$$

Equations [11.34] and [11.35] are valid when using the Levenberg-Marquardt method, but σ_ρ^2 is rather evaluated using the residual of $\|\mathbf{F}\|^2$:

$$\sigma_\rho^2 = \frac{\|\mathbf{F}(\mathbf{Q}_t, \mathbf{X}_t, \hat{\xi})\|^2}{(r - c)} \qquad [11.37]$$

In order to validate the success of the parameter estimation process, we can evaluate the residual error on some configurations that have not been used in the identification. We can also compare the values of the estimated parameters using different calibration methods.

11.5. Calibration methods

In this section, we present the most common calibration methods. The first method requires an external sensor, which provides either the location or the position coordinates of the end-effector; the second requires an external sensor to provide the distance traveled by the endpoint when moving from one configuration to another. The other methods are termed as autonomous because they only make use of the joint variables. They are based on realizing geometric constraints between the robot configurations or between the robot and the environment.

11.5.1. *Calibration using the end-effector coordinates*

This method is the most popular one and can be considered as the conventional approach. The function to be minimized is the difference between the measured and calculated end-effector locations. This method needs an external sensor to measure the location of the end-effector frame with respect to the world reference frame. In § 11.8, we describe some measurement systems that can be used for that. The nonlinear calibration model is given by:

$$ {}^{-1}\mathbf{T}_{n+1}(\mathbf{x}) - {}^{-1}\mathbf{T}_{n+1}(\mathbf{q}, \boldsymbol{\xi}) = \mathbf{0} \qquad [11.38] $$

where ${}^{-1}\mathbf{T}_{n+1}(\mathbf{x})$ is the measured location of the end-effector with respect to frame R_{-1}.

Equation [11.38] contains twelve elements that may be different from zero, but it has only six independent degrees of freedom. To obtain six independent elements, we rewrite this equation as:

$$ \Delta\mathbf{X} = \begin{bmatrix} \Delta\mathbf{X}_p(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi}) \\ \Delta\mathbf{X}_r(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi}) \end{bmatrix} = \mathbf{0} \qquad [11.39] $$

with:

- $\Delta\mathbf{X}_p$: (3x1) vector of the position error, equal to:

$$ \Delta\mathbf{X}_p = {}^{-1}\mathbf{P}_{n+1}(\mathbf{x}) - {}^{-1}\mathbf{P}_{n+1}(\mathbf{q}, \boldsymbol{\xi}) \qquad [11.40] $$

- $\Delta\mathbf{X}_r$: (3x1) vector of the orientation error (representing the difference between the measured and computed ${}^{-1}\mathbf{A}_{n+1}$), given by:

$$ \Delta\mathbf{X}_r = \mathbf{u}\,\alpha \qquad [11.41] $$

where $\mathbf{u}$ and α are obtained by solving the following equation (§ 2.3.7):

$$ {}^{-1}\mathbf{A}^{r}_{n+1} = \mathbf{rot}(\mathbf{u}, \alpha)\; {}^{-1}\mathbf{A}^{m}_{n+1} \qquad [11.42] $$

where ${}^{-1}\mathbf{A}^{r}_{n+1}(\mathbf{x}) = [\ \mathbf{s}_r \ \ \mathbf{n}_r \ \ \mathbf{a}_r\ ]$ is the measured (3x3) orientation matrix of frame R_{n+1}, and ${}^{-1}\mathbf{A}^{m}_{n+1}(\mathbf{q}, \boldsymbol{\xi}) = [\ \mathbf{s}_m \ \ \mathbf{n}_m \ \ \mathbf{a}_m\ ]$ is the computed orientation matrix using the direct geometric model.

If the orientation error is small, the following equation can be used (§ 2.3.8 and equation [2.35]):

$$ \Delta\mathbf{X}_r = \mathbf{u}\sin(\alpha) = \frac{1}{2}\begin{bmatrix} n_z - a_y \\ a_x - s_z \\ s_y - n_x \end{bmatrix} \qquad [11.43] $$

where the **s**, **n**, **a** components are obtained from the equation:

$$[\,\mathbf{s}\ \ \mathbf{n}\ \ \mathbf{a}\,] = {}^{-1}\mathbf{A}^{r}_{n+1}\,({}^{-1}\mathbf{A}^{m}_{n+1})^{-1} \qquad [11.44]$$

The linear differential model defining the deviation of the end-effector location due to the differential errors in the geometric parameters can be obtained as:

$$\Delta\mathbf{X} = \begin{bmatrix} \Delta\mathbf{X}_p(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi}) \\ \Delta\mathbf{X}_r(\mathbf{x}, \mathbf{q}, \boldsymbol{\xi}) \end{bmatrix} = \boldsymbol{\Psi}(\mathbf{q}, \boldsymbol{\xi})\, \Delta\boldsymbol{\xi} \qquad [11.45]$$

where:

- $\Delta\mathbf{X}$ represents the (6x1) vector of position and orientation errors (representing the difference between the measured and computed ${}^{-1}\mathbf{T}_{n+1}$);
- $\boldsymbol{\Psi}$ is the ($6xN_{par}$) generalized Jacobian matrix developed in § 11.3.

The calibration index of this method is 6. If we only measure the position of the endpoint, the first three equations of the nonlinear or linear calibration models only should be used. The calibration index reduces to 3.

It is worth noting that the geometric parameters have different units: meters (for distances), radians (for angles) or even no unit (for gear transmission ratios). The effect of this heterogeneity can be handled by introducing an appropriate weighting matrix. However, for industrial robots of about the size of a human arm, one obtains good results by using meters for the distances, radians for the angles, and by normalizing the encoder readings such that the elements K_i are of the order of one. Of course, if the links are much smaller (like fingers) or much larger (like excavators), the situation is different.

11.5.2. *Calibration using distance measurement*

In this method, we make use of the distance traveled by the endpoint when moving from one configuration to another [Goswami 93]. Thus, an external sensor measuring the distance such as an extendable ball bar system or a linear variable differential transformer (LVDT) is required. The calibration index of this method is 1. Let $D^{r}_{i,j}$ be the measured distance traveled by the endpoint between configurations $\mathbf{q}^i$ and $\mathbf{q}^j$. Thus, the nonlinear calibration equation is given by:

$$[P_x(\mathbf{q}^j,\boldsymbol{\xi}) - P_x(\mathbf{q}^i,\boldsymbol{\xi})]^2 + [P_y(\mathbf{q}^j,\boldsymbol{\xi}) - P_y(\mathbf{q}^i,\boldsymbol{\xi})]^2 + [P_z(\mathbf{q}^j,\boldsymbol{\xi}) - P_z(\mathbf{q}^i,\boldsymbol{\xi})]^2 = (D^{r}_{i,j})^2 \qquad [11.46]$$

The differential calibration model is given by:

$$2\{[P_x(q^j)-P_x(q^i)]\,[\Psi_x(q^j)-\Psi_x(q^i)]+[P_y(q^j)-P_y(q^i)]\,[\Psi_y(q^j)-\Psi_y(q^i)]+ \\ [P_z(q^j)-P_z(q^i)]\,[\Psi_z(q^j)-\Psi_z(q^i)]\}\,\Delta\xi = (D^r_{i,j})^2-(D_{i,j})^2 \quad [11.47]$$

where:

- $D_{i,j}$ is the computed distance traveled by the endpoint between configurations q^i and q^j, using the nominal parameters;
- Ψ_x, Ψ_y and Ψ_z denote the first, second and third rows of the generalized Jacobian matrix respectively.

11.5.3. *Calibration using location constraint and position constraint*

The main limitation of the previous approaches is that they require an accurate, fast and inexpensive external sensor to measure the Cartesian variables. Location constraint and position constraint methods are autonomous methods that do not require an external sensor. These methods can be used when the specified end-effector locations (or positions) can be realized by multiple configurations. It is worth noting that a robot with more than three degrees of freedom ($n > 3$) can be calibrated by the position constraint method, whereas for the location constraint method we must have $n \geq 6$ [Khalil 95b].

Let q^i and q^j represent two configurations giving the same location of the end-effector. Then, the nonlinear calibration model is given by:

$$^{-1}\mathbf{T}_{n+1}(q^j,\xi) - {}^{-1}\mathbf{T}_{n+1}(q^i,\xi) = \mathbf{0} \quad [11.48]$$

This equation can be transformed into a (6x1) vectorial equation as illustrated in § 11.5.1. The resulting equation is as follows:

$$\Delta X(q^i, q^j, \xi) = \begin{bmatrix} \Delta X_p(q^i, q^j, \xi) \\ \Delta X_r(q^i, q^j, \xi) \end{bmatrix} = 0 \quad [11.49]$$

The differential calibration model is given by:

$$\Delta X(q^i, q^j, \xi) = \left[\Psi(q^j,\xi) - \Psi(q^i,\xi)\right]\Delta\xi \quad [11.50]$$

where Ψ is the generalized Jacobian matrix developed in § 11.3, ΔX is the (6x1) vector representing the position and orientation differences between the locations $^{-1}\mathbf{T}_{n+1}(q^j,\xi)$ and $^{-1}\mathbf{T}_{n+1}(q^i,\xi)$.

The calibration index for this method is 6. In the position constraint method, the endpoint at configuration $\mathbf{q}^i$ should coincide with the endpoint at configuration $\mathbf{q}^j$. Thus, the first three equations of [11.49] and [11.50] only are considered. Hence, the calibration index is 3.

We note that the calibration Jacobian matrices of these methods are obtained by subtracting the generalized Jacobian matrices of two configurations. Thus, the number of identifiable parameters is less than that of the conventional methods (Example 11.1). For example:

- if a column of the generalized Jacobian matrix is constant, then the corresponding parameter will not be identified. This is the case of the parameters $\Delta\gamma_z$, Δb_z, $\Delta\alpha_z$, Δd_z, $\Delta\theta_{1'}$, $\Delta r_{1'}$, $\Delta\beta_0$ for both location constraint and position constraint methods;
- if a column of the generalized Jacobian matrix is identical for any two configurations $\mathbf{q}^i$ and $\mathbf{q}^j$ satisfying the calibration constraint, then the corresponding parameter cannot be identified. For example, since in the location constraint method ${}^{-1}\mathbf{T}_{n+1}(\mathbf{q}^i) = {}^{-1}\mathbf{T}_{n+1}(\mathbf{q}^j)$, then ${}^{-1}\mathbf{T}_n(\mathbf{q}^i) = {}^{-1}\mathbf{T}_n(\mathbf{q}^j)$. Thus, the parameters $\Delta r_{n'}$, $\Delta\theta_{n'}$, $\Delta\beta_{n'}$, Δd_e, $\Delta\alpha_e$, $\Delta\theta_e$, Δr_e are not identifiable.

11.5.4. *Calibration methods using plane constraint*

In this approach, the calibration is carried out using the values of the joint variables of a set of configurations for which the endpoint of the robot is constrained to lie in the same plane. Several methods based on this technique have been proposed [Tang 94], [Zhong 95], [Khalil 96b], [Ikits 97]. The main advantage of this autonomous method is the possibility to collect the calibration points automatically using touch or tactile sensor (such as LVDT, a trigger probe, or a laser telemeter). Two methods are developed in this section: the first makes use of the plane equation while the second uses the coordinates of the normal to the plane. The calibration index of these methods is 1. Special care has to be taken to locate the constraint plane and to select the calibration points in order to obtain a good condition number for the observation matrix [Ikits 97], [Khalil 00b].

11.5.4.1. *Calibration using plane equation*

Since the endpoints are in the same plane, and assuming that the plane does not intersect the origin, the nonlinear calibration model is:

$$aP_x(\mathbf{q}, \xi) + bP_y(\mathbf{q}, \xi) + cP_z(\mathbf{q}, \xi) + 1 = 0 \qquad [11.51]$$

where:

- a, b, c represent the plane coefficients referred to the reference world frame;
- P_x, P_y, P_z represent the Cartesian coordinates of the endpoint relative to the reference frame.

Applying equation [11.51] to a sufficient number of configurations, the resulting system of nonlinear equations can be solved to estimate the plane coefficients and the identifiable geometric parameters.

Using a first order development for equation [11.51] leads to the following linearized calibration model:

$$\begin{bmatrix} P_x(\mathbf{q}) & P_y(\mathbf{q}) & P_z(\mathbf{q}) & a\Psi_x(\mathbf{q})+b\Psi_y(\mathbf{q})+c\Psi_z(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta c \\ \Delta\xi \end{bmatrix} = -aP_x(\mathbf{q}) - bP_y(\mathbf{q}) - cP_z(\mathbf{q}) - 1 \quad [11.52]$$

where:

- Ψ_x, Ψ_y and Ψ_z are the first, second and third rows of the generalized Jacobian matrix respectively;
- $P_x(\mathbf{q})$, $P_y(\mathbf{q})$, $P_z(\mathbf{q})$ represent the computed Cartesian coordinates of the endpoint in the reference frame.

The coefficients of the plane are initialized by solving the equation of the plane for the collected configurations:

$$\begin{bmatrix} -1 \\ \cdots \\ -1 \end{bmatrix} = \begin{bmatrix} P_x^1 & P_y^1 & P_z^1 \\ \cdots & \cdots & \cdots \\ P_x^e & P_y^e & P_z^e \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \rho \quad [11.53]$$

where P_x^j, P_y^j and P_z^j are the coordinates of the endpoint as given by the DGM with the nominal values of the geometric parameters for configuration $\mathbf{q}^j$.

If the coefficients a, b, and c of the plane are known, the corresponding columns and unknowns in equation [11.52] are eliminated. The resulting linear calibration model is given by:

$$[a\Psi_x(\mathbf{q}) + b\Psi_y(\mathbf{q}) + c\Psi_z(\mathbf{q})]\,\Delta\xi = -aP_x(\mathbf{q}) - bP_y(\mathbf{q}) - cP_z(\mathbf{q}) - 1 \quad [11.54]$$

11.5.4.2. *Calibration using normal coordinates to the plane*

In this method, the calibration model is established by using the fact that the scalar product of the vector normal to the plane and the vector between two points in the plane is zero [Zhong 95], [Maurine 96]. The coordinates of the normal can be obtained with inclinometers. Since this method is independent of the plane position, it may give poor results if the initial values of the robot parameters are not close to the real values.

The nonlinear calibration model for the configurations $\mathbf{q}^i$ and $\mathbf{q}^j$ is such that:

$$a[P_x(\mathbf{q}^j, \xi) - P_x(\mathbf{q}^i, \xi)] + b[P_y(\mathbf{q}^j, \xi) - P_y(\mathbf{q}^i, \xi)] + c[P_z(\mathbf{q}^j, \xi) - P_z(\mathbf{q}^i, \xi)] = 0 \quad [11.55]$$

Assuming that the normal coordinates are known, we obtain the following linearized equation:

$$\left\{ a[\Psi_x(\mathbf{q}^j) - \Psi_x(\mathbf{q}^i)] + b[\Psi_y(\mathbf{q}^j) - \Psi_y(\mathbf{q}^i)] + c[\Psi_z(\mathbf{q}^j) - \Psi_z(\mathbf{q}^i)] \right\} \Delta\xi = -a[P_x(\mathbf{q}^j) - P_x(\mathbf{q}^i)] - b[P_y(\mathbf{q}^j) - P_y(\mathbf{q}^i)] - c[P_z(\mathbf{q}^j) - P_z(\mathbf{q}^i)] \quad [11.56]$$

where Ψ_u, $\mathbf{P}_u$, for $u = x, y, z$, are computed in terms of $\mathbf{q}$ and ξ.

NOTE.– The concatenation of the equations of several planes in a unique system of equations increases the number of identifiable parameters [Zhuang 99]. The use of three planes gives the same identifiable parameters as in the position measurement method if the robot has at least one prismatic joint, while four planes are needed if the robot has only revolute joints [Besnard 00b].

• **Example 11.1.** Determination of the identifiable parameters with the previous methods for the Stäubli RX-90 robot (Figure 3.3b) and the Stanford robot (Figure 11.2). The geometric parameters of these robots are given in Tables 11.1 and 11.2 respectively. For the plane constraint methods, we assume that the plane coefficients are known.

Tables 11.3 and 11.4 present the identifiable geometric parameters of the two robots as provided by the software package GECARO "GEometric CAlibration of Robots" [Khalil 99a], [Khalil 00b]. The parameters indicated by "0" are not identifiable, because they have no effect on the identification model, while the parameters indicated by "n" are not identified because they have been grouped with some other parameters.

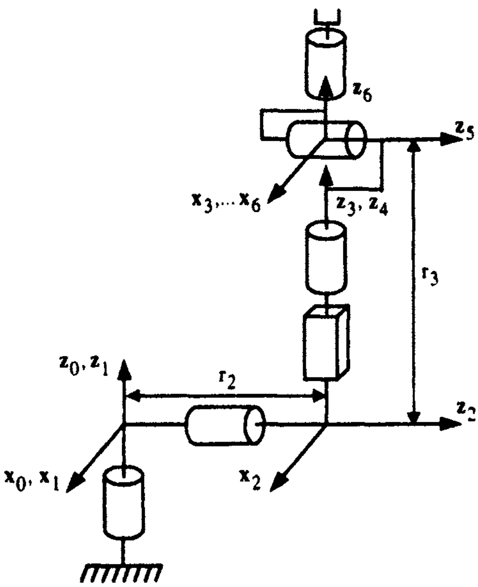

Figure 11.2. *Stanford robot*

Table 11.1. *Geometric parameters of the RX-90 Stäubli robot*[1]

j	σ_j	α_j	d_j	θ_j	r_j	β_j	K_j
0	2	0	0	$\pi/2$	0.5	0	0
1	0	0.1	0	θ_1	0	0	1
2	0	$-\pi/2$	0	θ_2	0	0	1
3	0	0	0.5	θ_3	0	0	1
4	0	$\pi/2$	0	θ_4	0.5	0	1
5	0	$-\pi/2$	0	θ_5	0	0	1
6	0	$\pi/2$	0	θ_6	0.3[2]	0	1
7	2	1.3	0.2	$\pi/2$	0.1	0	0

[1] Distances are in meters and angles are in radians.
[2] r_6 is equal to the end-effector frame parameter b_e.

Table 11.2. *Geometric parameters of the Stanford robot*[3]

j	σ_j	α_j	d_j	θ_j	r_j	β_j	K_j
0	2	0	0	$\pi/2$	0.5	0	0
1	0	0.1	0	θ_1	0	0	1
2	0	$-\pi/2$	0	θ_2	0.2	0	1
3	1	$\pi/2$	0	0	r_3	0	1
4	0	0	0	θ_4	0	0	1
5	0	$-\pi/2$	0	θ_5	0	0	1
6	0	$\pi/2$	0	θ_6	0.3[1]	0	1
7	2	1.3	0.2	$\pi/2$	0.1	0	0

From Tables 11.3 and 11.4, the following general results are deduced:

1) the location measurement method allows the identification of the maximum number of parameters (36 parameters for the Stäubli RX-90 robot and 34 for the Stanford robot), which corresponds to the following general equation [Everett 88]:

$$b = 4(n_r + 1) + 2 + 2n_p + n$$

including:

- $4(n_r + 1)$ parameters for the n_r revolute joints and for frame R_{n+1};
- 2 parameters for R_0;
- 2 n_p parameters for the prismatic joints;
- n parameters for the joint gear transmission ratios;

2) the parameters of frames R_0 and R_1 have no effect on the model and cannot be identified when using the following methods: distance measurement, position constraint and location constraint;

3) most of the parameters of frames R_n and R_{n+1}, (R_6 and R_7), are not identifiable with the location constraint method;

4) most of the parameters of frames R_0, R_1 and R_7 are not identifiable with the planar methods. Some of them are grouped with other parameters;

5) the parameter β_j is not identifiable when $\alpha_j \neq 0$;

6) the offsets of joint variables 2, ..., $n-1$ are identifiable with all the methods;

---

[3] Distances are in meters and angles are in radians.

7) the offset of joint 1 is not identifiable with the following methods: distance measurement, position constraint, location constraint; whereas the offset of joint n is not identifiable with the location constraint method;

8) all the gear transmission ratios K_j are identifiable;

9) the parameter r_6 is not identifiable with the position constraint and the plane constraint methods for both robots. It represents the scale factor of these methods. Note that the constraint equations are also verified when all the distances are zero.

10) in the location constraint method, the parameter r_6 has no effect. The scale factor is represented by r_4 for the Stäubli robot and r_2 for the Stanford robot. It is worth noting that in the case of the Stanford robot, the scale factor could be the prismatic variable r_3 (instead of r_6 or r_2) if we assume that the gear transmission ratio K_3 is known and has not to be identified;

11) the parameters α_7, θ_7 and r_7 are not identifiable with the position measurement, position constraint, and plane constraint methods. This is because the end-effector reduces to a point that is defined by the three parameters r_6, θ_6 and d_7.

11.6. Correction and compensation of errors

Once the identification is performed, the estimated parameters must be integrated into the robot controller. Computing the DGM with the general identified parameters is more time consuming than the analytical solution, but it can be performed on-line. If the IGM were computed using a general numerical iterative algorithm, we could compensate all the calibrated parameters. But, a major problem stems for industrial robots whose IGM is analytically implemented. The calibration may result in a non-analytically solvable robot. For example, a spherical wrist could be found not to be spherical. In this case, the following geometric parameters can be updated in the controller straightaway: end-effector parameters, robot base parameters, joint offsets, r_j and d_j whose nominal values are not zero, and the angles α_j, θ_j whose nominal values are not k $\pi/2$, with k being an integer. For the other parameters, an iterative approach must be implemented. A possible approach is to use the closed-form solution in order to compute a good first guess. Then, an accurate solution is obtained using an iterative algorithm. Such a process converges in a small number of iterations, or even in a single iteration if the end-effector location error due to the geometric parameters to be compensated is not too high. The iterative tuning process is summarized as follows (Figure 11.3):

Table 11.3. *Identifiable parameters of the RX-90 Stäubli robot*

Parameter	Position measurement	Location measurement	Distance measurement	Position constraint	Location constraint	Plane equation	Plane normal
θ0			0	0	0	n	n
r0			0	0	0		0
α1			0	0	0		
d1			0	0	0	n	0
θ1			0	0	0		
r1			0	0	0	n	0
β1	n	n	0	0	0	n	n
α2							
d2							
θ2							
r2							
β2	n	n	n	n	n	n	n
α3							
d3							
θ3							
r3	n	n	n	n	n	n	n
β3							
α4							
d4							
θ4							
r4					n		
β4	n	n	n	n	n	n	n
α5							
d5							
θ5							
r5							
β5	n	n	n	n	n	n	n
α6							
d6							
θ6					0		
r6				n	0	n	n
β6	n	n	n	n	0	n	n
α7	n		n	n	0	n	n
d7					0		
θ7	0		0	0	0	0	0
r7	n		n	n	0	n	n
β7	n	n	n	n	0	n	n
K1,...K6							
Total	**33**	**36**	**27**	**26**	**23**	**29**	**28**

(n: non-identifiable parameter. Its effect is grouped with some other parameters.
0: non-identifiable parameter having no effect on the model)

Table 11.4. *Identifiable parameters of the Stanford robot*

Parameter	Position measurement	Location measurement	Distance measurement	Position constraint	Location constraint	Plane equation	Plane normal
$\theta0$			0	0	0	n	n
r0			0	0	0		0
$\alpha1$			0	0	0		
d1			0	0	0	n	0
$\theta1$			0	0	0		
r1			0	0	0	n	0
$\beta1$	n	n	0	0	0	n	n
$\alpha2$							
d2							
$\theta2$							
r2					n		
$\beta2$	n	n	n	n	n	n	n
$\alpha3$							
d3							
$\theta3$	n	n	n	n	n	n	n
r3							
$\beta3$	n	n	n	n	n	n	n
$\alpha4$							
d4	n	n	n	n	n	n	n
$\theta4$							
r4	n	n	n	n	n	n	n
$\beta4$							
$\alpha5$							
d5							
$\theta5$							
r5							
$\beta5$	n	n	n	n	n	n	n
$\alpha6$							
d6							
$\theta6$					0		
r6				n	0	n	n
$\beta6$	n	n	n	n	0	n	n
$\alpha7$	n		n	n	0	n	n
d7					0		
$\theta7$	0		0	0	0	0	0
r7	n		n	n	0	n	n
$\beta7$	n	n	n	n	0	n	n
K1, ...K6							
Total	**31**	**34**	**25**	**24**	**21**	**27**	**26**

(n: non-identifiable parameter. Its effect is grouped with some other parameters.
0: non-identifiable parameter having no effect on the model)

1) use the analytical IGM to compute the joint variables **q** corresponding to the desired end-effector location $^{-1}\mathbf{T}^{d}_{n+1}$;
2) compute the differential error $\Delta\mathbf{X}$ between $^{-1}\mathbf{T}^{d}_{n+1}$ and $^{-1}\mathbf{T}^{c}_{n+1}(\hat{\xi}, \mathbf{q})$, where $^{-1}\mathbf{T}^{c}_{n+1}(\hat{\xi}, \mathbf{q})$ indicates the direct geometric model using the estimated parameters. Note that $\Delta\mathbf{X}$ can also be computed using the generalized differential model (§ 11.3);
3) if $\Delta\mathbf{X}$ is sufficiently small, stop the tuning process. Otherwise, compute $\Delta\mathbf{q}$ corresponding to the error $\Delta\mathbf{X}$ using the classical inverse differential model $\Delta\mathbf{q} = \mathbf{J}^{+}\Delta\mathbf{X}$;
4) update the joint variables such that $\mathbf{q} = \mathbf{q} + \Delta\mathbf{q}$;
5) return to step 2.

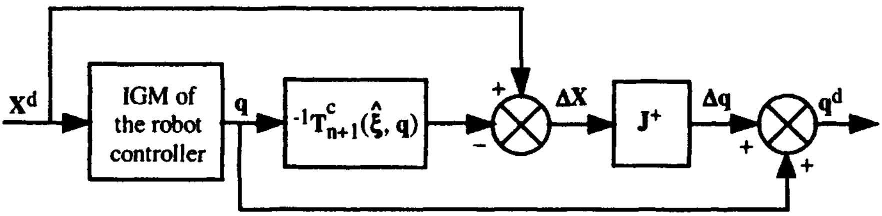

Figure 11.3. *Principle of compensation*

In the context of off-line programming systems, once a calibration has been performed, the compensated joint values can be downloaded directly to the controller.

11.7. Calibration of parallel robots

One of the attractive feature of parallel robots is their potential for higher accuracy as compared to serial robots, mainly due to the higher stiffness of their closed-loop structure. However, this stiffness does not result directly into better accuracy, but rather into higher repeatability. Therefore, good calibration of the geometric parameters is also necessary for a parallel robot to improve its accuracy.

In this section, we consider the case of a six degree-of-freedom parallel robot of the Gough-Stewart family (Figures 8.1 and 8.11). It is composed of six legs of (RR)-P-(RRR) architecture, a fixed base, and a mobile platform to which the tool is attached. The prismatic joints are actuated, while the universal joints (U-joints) and the spherical joints (S-joints) are passive. The reference frame R_f is assumed to be attached to the base and the end-effector frame R_E is attached to the platform. We assume that the universal and spherical joints are perfect, and that the prismatic joints are perfectly assembled. Consequently, 42 geometric parameters are needed to compute the geometric model, namely: six leg lengths denoted by the joint variable vector $\mathbf{q}$, 3x6 coordinates of the centers of the U-joints with respect to the reference frame (${}^{f}\mathbf{P}_{Ai}$, for i = 1, ..., 6), and 3x6 coordinates of the centers of the S-joints with respect to the end-effector frame (${}^{E}\mathbf{P}_{Bi}$, for i = 1, ..., 6). The geometric calibration consists of estimating these parameters accurately. As in serial robots, the calibration procedure consists of four steps: construction of a calibration model, collection of a sufficient number of configurations, identification of the geometric parameters from the calibration equation, and implementation of error compensation. It is worth noting that for parallel robots the parameters estimated can be introduced into the controller straightaway.

Various calibration models are proposed in the literature. We will only consider the conventional techniques where the calibration model is a function of the Cartesian coordinates of the location of the mobile platform frame, which must be provided by an external sensor. In this case, the calibration model can be derived from the IGM [Zhuang 95] or the DGM. Recall that the IGM is easy to compute using closed-form expressions and that it gives a unique solution $\mathbf{q}$ for a desired location ${}^{f}\mathbf{T}_{E}$, whereas the DGM has multiple solutions and is computed iteratively.

11.7.1. *IGM calibration model*

The error function in the IGM calibration model is the difference between the measured and computed joint variables. It is represented by the following equation:

$$\mathbf{q} - \mathrm{IGM}({}^{f}\mathbf{T}_{E}(\mathbf{x}), \boldsymbol{\xi}) = \mathbf{0} \qquad [11.57]$$

where $\boldsymbol{\xi}$ denotes the current geometric parameters, and ${}^{f}\mathbf{T}_{E}(\mathbf{x})$ is the measured location of the end-effector frame relative to the fixed reference frame.

The IGM providing the joint position q_i in terms of the desired transformation matrix ${}^{f}\mathbf{T}_{E}$ and of the fixed geometric parameters ${}^{f}\mathbf{P}_{Ai}$ and ${}^{E}\mathbf{P}_{Bi}$ is given by equation [8.11]. It is rewritten as:

$$q_i^2 = ({}^{f}\mathbf{P}_{Bi} - {}^{f}\mathbf{P}_{Ai})^T ({}^{f}\mathbf{P}_{Bi} - {}^{f}\mathbf{P}_{Ai}) \qquad [11.58]$$

The joint variable q_i is given by:

$$q_i = D_i + q_{ci} \qquad [11.59]$$

where q_{ci} represents the joint i sensor value and D_i is a constant offset.

The nonlinear calibration model is obtained from equation [11.58] as:

$$q_i^2 - ({}^f\mathbf{A}_E\,{}^E\mathbf{P}_{Bi} + {}^f\mathbf{P}_E - {}^f\mathbf{P}_{Ai})^T ({}^f\mathbf{A}_E\,{}^E\mathbf{P}_{Bi} + {}^f\mathbf{P}_E - {}^f\mathbf{P}_{Ai}) = 0 \qquad [11.60]$$

Equation [11.60] shows that the parameters of each leg can be identified separately. Thus, we can split the identification problem into six independent systems of equations where each system is a function of the seven parameters of one leg. Applying equation [11.60] for a sufficient number of configurations e and concatenating all the equations together leads to the following nonlinear equation:

$$\mathbf{0} = \mathbf{F}_i(\mathbf{Q}_{ti}, \mathbf{X}_t, \boldsymbol{\xi}_i) + \boldsymbol{\rho}_i \qquad [11.61]$$

where $\mathbf{Q}_{ti} = q_i^1, \ldots, q_i^e$ represents the calibration configurations of leg i, $\mathbf{X}_t = \mathbf{x}^1, \ldots, \mathbf{x}^e$ indicates the corresponding locations of the mobile platform, $\boldsymbol{\xi}_i$ indicates the geometric parameters of leg i, and $\boldsymbol{\rho}_i$ is the vector of modeling error for leg i.

This nonlinear optimization problem can be solved by the Levenberg-Marquardt algorithm as described in § 11.4.3.

The Jacobian matrix of the IGM calibration method is formulated for leg i by:

$$\Delta q_i = \boldsymbol{\phi}_i(\mathbf{x}, \boldsymbol{\xi}_i)\, \Delta\boldsymbol{\xi}_i \qquad [11.62]$$

The closed-form expressions of the columns of the calibration Jacobian matrix $\boldsymbol{\phi}_i$ can be obtained by differentiating equations [11.58] and [11.59] with respect to the elements of ${}^E\mathbf{P}_{Bi}$, ${}^f\mathbf{P}_{Ai}$ and D_i:

$$\Delta q_i = \frac{1}{q_i} \begin{bmatrix} (-{}^f\mathbf{P}_{Bi} + {}^f\mathbf{P}_{Ai})^T & ({}^f\mathbf{P}_{Bi} - {}^f\mathbf{P}_{Ai})^T\,{}^f\mathbf{A}_E & q_i \end{bmatrix} \begin{bmatrix} \Delta{}^f\mathbf{P}_{Ai} \\ \Delta{}^E\mathbf{P}_{Bi} \\ \Delta D_i \end{bmatrix} \qquad [11.63]$$

Since the Jacobian matrix can be obtained using closed-form expressions, we can use the iterative pseudoinverse solution to solve this problem as for serial robots. The prediction error function of all the joints at configuration $\mathbf{q}^j$ is calculated in terms of the measured location ${}^f\mathbf{T}_E^j$ and the current geometric parameters $\boldsymbol{\xi}$ by:

$$\Delta\mathbf{q}^j = \mathbf{q}^j - \mathrm{IGM}({}^f\mathbf{T}_E^j, \boldsymbol{\xi}) \qquad [11.64]$$

11.7.2. *DGM calibration model*

The second nonlinear calibration model, which makes use of the iterative DGM, is given by:

$$\Delta\mathbf{X}(\mathbf{q}, {}^{f}\mathbf{T}_{E}(\mathbf{x}), \boldsymbol{\xi}) = \mathbf{0} \qquad [11.65]$$

where $\Delta\mathbf{X}$ indicates the (6x1) vector of the difference between the measured location ${}^{f}\mathbf{T}_{E}(\mathbf{x})$ and the computed location of the platform $DGM(\mathbf{q}^{j}, \boldsymbol{\xi})$.

The vector $\Delta\mathbf{X}$ can be determined as given in § 11.5.1 by computing $\Delta\mathbf{X}_{p}$ and $\Delta\mathbf{X}_{r}$. The identification equation can be solved using the Levenberg-Marquardt method.

The advantage of the DGM calibration approach is the possibility to use in the calibration equation partial elements of the location of the end-effector [Besnard 99]. For example, we can carry out the calibration using the position coordinates of the platform by considering the first three components of equation [11.65]. In the same manner, we can also make use of two inclinometers [Besnard 99]. By comparison, the IGM calibration approach needs complete measurement of the location. The main drawback of the DGM method is its computational complexity. Besides, the corresponding Jacobian matrix cannot be computed analytically.

Before closing this section, we have to mention the autonomous calibration methods that are based on measuring some of the passive joint variables [Zhuang 97] or by locking some of them [Murareci 97], [Besnard 00a], [Daney 00], [Khalil 99b]. Recall that providing some passive joints with sensors simplifies the direct geometric model solution. With these autonomous methods, the coordinates of points A_i and B_i, for i = 1, ..., 6, are estimated with respect to frames R_0 and R_m respectively and not with respect to frames R_f and R_E (see § 8.6.1 for the definition of these frames). The identifiable parameters of all these methods are presented in [Besnard 01].

11.8. Measurement techniques for robot calibration

Conventional calibration methods, as well as the evaluation of positioning accuracy and repeatability of robots, requires measurement of either the end-effector location or position with high accuracy. Most of the current measurement schemes are based on vision systems, measuring machines, laser interferometers, laser tracking systems, and theodolites [Mooring 91], [Bernhardt 93].

Ideally, the measurement system should be accurate, inexpensive and should be operated automatically. The goal is to minimize the calibration time and the robot unavailability. At this time, such devices are not yet available. Nevertheless, we present in this section some principles that have given place to industrial realization.

11.8.1. *Three-cable system*

Such a system is basically composed of three high resolution optical encoders P1, P2, P3. Low mass cables are fixed to one of their ends on the encoder shafts whereas the other ends are fixed on the endpoint M of the robot (Figure 11.4). The encoder readings give the cable lengths, which represent the radii ρ_1, ρ_2, ρ_3 of three spheres whose centers are on the encoder shafts. The intersection of the spheres determines the coordinates of M.

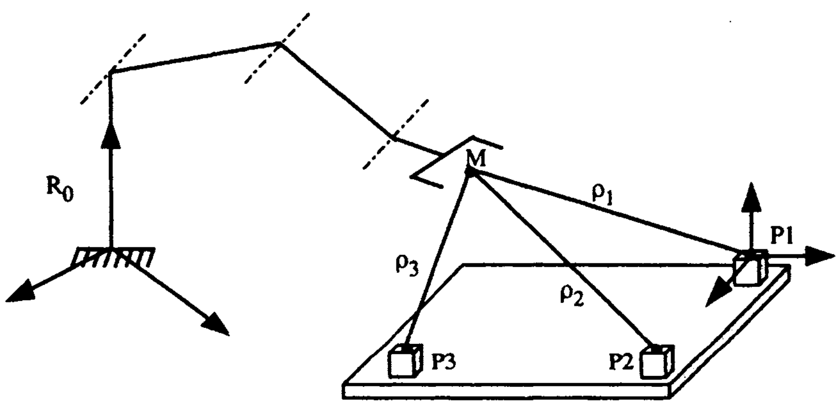

Figure 11.4. *A three-cable system*

This low cost device provides automatically the coordinates of the endpoint M. As a commercial example of such a system, we can mention the *3D CompuGauge* from Dynalog whose accuracy is about 0.1 mm for a cubic measuring space of 1.5 m of side.

11.8.2. *Theodolites*

A theodolite is a telescope where the two angles giving the orientation of the line of sight can be measured precisely. The Cartesian coordinates of a target ball M on the end-effector can be obtained in terms of the readings of two theodolites Th1 and Th2 pointing this ball and of the transformation matrix **T** between the two theodolites (Figure 11.5).

The accuracy of this system is excellent (of the order of 0.02 mm at about 1 m distance). The cost of a theodolite is rather high. The first systems had to be manually operated, but the data were read and stored by a computer (ECDS3 from Leica for example). Now, we can find motorized theodolites that can track automatically an illuminated target ball (TCA from Leica, Elta from Zeiss). Their cost is naturally greater.

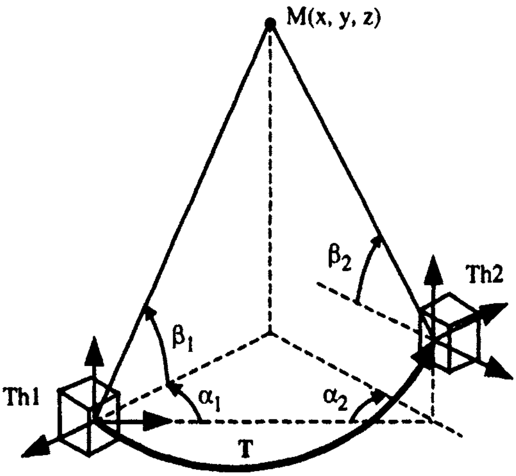

Figure 11.5. *Measurement system using two theodolites*

11.8.3. *Laser tracking system*

This device is composed of two 2-D scanner systems T1 and T2 external to the robot. Each of the two systems deflects a laser beam in a vertical plane and a horizontal plane, thanks to two motorized mirrors. The direction of the beam is controlled to track a retroreflector fixed on the terminal link of the robot (Figure 11.6). The position of the target point is calculated automatically using the angles of the laser beams. The precision is of the order of 0.1 mm for a target at 1 m distance. As examples of this system, we can mention the LASERTRACES system from ASL (UK) and OPTOTRAC from the University of Surrey (UK). The limitation of this system is the requirement of a dedicated end-effector.

11.8.4. *Camera-type devices*

The principle is to acquire at least simultaneously two images of the robot configuration using two cameras. The two images are processed in real time to estimate the 3D coordinates of target markers attached to the robot links.

Practically, the existing systems differ by the number and the type of the cameras used. We can mention as an example the RODYM6D from Krypton, which uses an OPTOTRAK sensor from Northern Digital. The sensor is composed of 3 CCD cameras and can handle up to 24 infrared light emitting diode markers. The precision is of the order of 0.2 mm for points at 2.5 m distance.

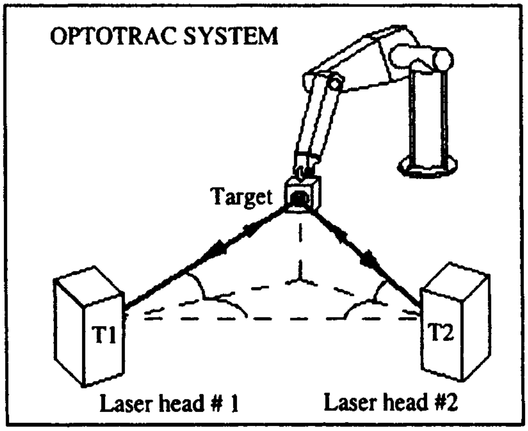

Figure 11.6. *Laser tracking system (University of Surrey)*

11.9. Conclusion

We have presented various approaches for the geometric calibration of serial robots. The geometric parameters of the robot, the base frame parameters and the end-effector frame parameters are defined using the Hayati modification of Khalil-Kleinfinger notations. All of the calibration methods are described by a unified nonlinear equation and a general linear equation. The Jacobian matrix of each calibration method is obtained as a function of the generalized Jacobian matrix relating the variation of the end-effector location with the geometric parameter variation. The generalized Jacobian matrix is computed using an efficient method making use of the elements of the transformation matrices of the link frames. The identifiable parameters are determined numerically by studying the QR decomposition of the observation matrix using random configurations satisfying the constraints of the calibration method. The nonlinear estimation problem is resolved using the Levenberg-Marquardt method or using an iterative pseudoinverse method. The optimum selection of the calibration configurations is treated by minimizing the condition number of the observation matrix. These methods can be extended to include the calibration of joint elasticity and link flexibility [Besnard 00a].

We have also presented the geometric calibration of parallel robots when the measurement of the end-effector location is available. The problem can be formulated either with inverse or with direct geometric models. References are given for autonomous calibration approaches for these structures.

The instrumentation for geometric calibration is quite varied. Good commercial systems measuring the end-effector coordinates are beginning to emerge, and some of them have been described. The autonomous calibration methods do not need such external sensors. They are only based on the joint sensor readings, which are available with good precision on all the robots.

In the next chapter, we address the problem of estimating the inertial parameters and friction parameters.