14. Model Identification

John Hollerbach, Wisama Khalil, Maxime Gautier

This chapter discusses how to determine the kinematic parameters and the inertial parameters of robot manipulators. Both instances of model identification are cast into a common framework of least-squares parameter estimation, and are shown to have common numerical issues relating to the identifiability of parameters, adequacy of the measurement sets, and numerical robustness. These discussions are generic to any parameter estimation problem, and can be applied in other contexts.

For kinematic calibration, the main aim is to identify the geometric Denavit–Hartenberg (DH) parameters, although joint-based parameters relating to the sensing and transmission elements can also be identified. Endpoint sensing or endpoint constraints can provide equivalent calibration equations. By casting all calibration methods as closed-loop calibration, the calibration index categorizes methods in terms of how many equations per pose are generated.

Inertial parameters may be estimated through the execution of a trajectory while sensing one or more components of force/torque at a joint. Load estimation of a handheld object is simplest because of full mobility and full wrist force-torque sensing. For link inertial parameter estimation, restricted mobility of links nearer the base as well as sensing only the joint torque means that not all inertial parameters can be identified. Those that can be identified are those that affect joint torque, although they may appear in complicated linear combinations.

14.1 Overview

There are many different kinds of models in robotics, whose accurate identification is required for precise control. Examples from the previous chapters include sensor models, actuator models, kinematic models, dynamic models, and flexibility models. System identification is the general field concerned with the process of identifying models from measurements. Generally speaking, there are two types of models: parametric and nonparametric models. Parametric models are described by a few parameters, which are adequate to characterize the accuracy of a model throughout the working range. Examples include sensor gain and offset, the DH parameters for links, and rigid-body inertial parameters. Parametric models are particularly appropriate for robotics, whose components are manmade and whose properties are controlled and understood.

Nonparametric models include the impulse response and the Bode plot for linear systems, and Wiener and Volterra kernels for nonlinear systems [14.1]. A nonparametric model can be used as a stepping stone towards identifying a parametric model; for example, a Bode plot (graph of amplitude and phase versus input frequency) is often used to decide on model order, such as whether an actuator should be modeled as a second- or third-order system. Otherwise, nonparametric models are required when the system's properties are so complicated that a few lumped parameters do not suffice. This is particularly true for biological systems.

This chapter will describe the parametric calibration of the following kinds of models.

1. Kinematic parameters. Kinematic calibration is the process of locating coordinate systems of objects relative to each other. These objects may be isolated from each other, or they may be linked by joints. Examples include
   - locating a robot relative to a global reference frame
   - locating a stereo vision system relative to a robot
   - locating a grasped object relative to the manipulator's gripper frame
   - locating neighboring coordinate systems of links of a robot manipulator
2. Rigid-body inertial parameters. These parameters are required to predict the driving forces and torques to move objects and manipulators.

Suppose there are N_{par} parameters combined into an $N_{\text{par}} \times 1$ parameter vector $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_{N_{\text{par}}}\}$. The parameters may appear linearly or nonlinearly in the model.

$$\text{Linear model:} \quad \boldsymbol{y}^l = \boldsymbol{A}^l \boldsymbol{\phi} \,, \tag{14.1}$$

$$\text{Nonlinear model:} \quad \boldsymbol{y}^l = \boldsymbol{f}(\boldsymbol{x}^l, \boldsymbol{\phi}) \,, \tag{14.2}$$

where $\boldsymbol{y}^l = \{y_1^l, \ldots, y_M^l\}$ is the $M \times 1$ vector of output variables and $\boldsymbol{x}^l = \{x_1^l, \ldots, x_n^l\}$ is the vector of input variables. For the linear model, $\boldsymbol{A}^l$ is an $M \times N_{\text{par}}$ matrix whose entries A_{ij}^l are functions of the input variables $\boldsymbol{x}^l$. Any entry A_{ij}^l may be a complicated nonlinear function of $\boldsymbol{x}^l$, but it evaluates to just a number. For the nonlinear model, an explicit equation is shown in which the input variables appear in a nonlinear function $\boldsymbol{f} = \{f_1, \ldots, f_M\}$. Implicit nonlinear models $\boldsymbol{f}(\boldsymbol{y}^l, \boldsymbol{x}^l, \boldsymbol{\phi}) = \boldsymbol{0}$ may also appear in calibration [14.2]; they are handled similarly to explicit nonlinear models (Sect. 14.2.2). There may be P different measurements; a particular measurement is indicated by the superscript $l = 1, \ldots, P$.

For linear models, information from different measurements is combined by stacking the P equations (14.1):

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\phi} \,, \tag{14.3}$$

where $\boldsymbol{y} = \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^l\}$ is an $MP \times 1$ vector of all output measurements and $\boldsymbol{A} = \{\boldsymbol{A}^1, \ldots, \boldsymbol{A}^l\}$ is $MP \times N_{\text{par}}$. The parameters are estimated by ordinary least squares:

$$\boldsymbol{\phi} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{y} \,. \tag{14.4}$$

In statistics, the matrix $\boldsymbol{A}$ is called the regressor matrix and the least-squares solution is called regression [14.3]. An example of a linear model is the rigid-body model for inertial parameters.

The Gauss–Newton method [14.3] is typically employed to estimate the nonlinear model (14.2). First, absorb the input variables $\boldsymbol{x}^l$ (which can be considered as a number of constants) into the nonlinear function $\boldsymbol{f}^l$, now given a superscript. The model is linearized through a Taylor series expansion at a current estimate $\boldsymbol{\phi}^k$ of the parameters at iteration k:

$$\begin{aligned}
\boldsymbol{y}_{\text{c}}^l &= \boldsymbol{f}^l(\boldsymbol{\phi}^k + \Delta\boldsymbol{\phi}) \\
&= \boldsymbol{f}^l(\boldsymbol{\phi}^k) + \left.\frac{\partial \boldsymbol{f}^l(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}^k} \Delta\boldsymbol{\phi} \\
&\quad + \text{higher order terms} \\
&\approx \boldsymbol{f}^l(\boldsymbol{\phi}^k) + \boldsymbol{A}^l \Delta\boldsymbol{\phi} \,,
\end{aligned} \tag{14.5}$$

where $\boldsymbol{y}_{\text{c}}^l$ are the computed values of the output variables and $\boldsymbol{A}^l = \partial \boldsymbol{f}^l / \partial \boldsymbol{\phi}$ is a Jacobian matrix evaluated at $\boldsymbol{\phi}^k$. Higher-order terms in the Taylor series are ignored, yielding the linearized form (14.5). The bold assumption is now made that a correction $\Delta\boldsymbol{\phi}$ to the parameter estimate $\boldsymbol{\phi}^k$ causes the computed output variables to equal the measurements: $\boldsymbol{y}_{\text{c}}^l = \boldsymbol{y}^l$. Defining $\Delta\boldsymbol{y}^l = \boldsymbol{y}^l - \boldsymbol{f}^l(\boldsymbol{\phi}^k)$ as the error between the output measurement and the predicted output with the current model $\boldsymbol{\phi}^k$, the linearized equation (14.5) becomes

$$\Delta\boldsymbol{y}^l = \boldsymbol{A}^l \Delta\boldsymbol{\phi} \,. \tag{14.6}$$

The linearized equation is then stacked for P measurements for the estimation form:

$$\Delta\boldsymbol{y} = \boldsymbol{A}\Delta\boldsymbol{\phi} \,. \tag{14.7}$$

A correction $\Delta\boldsymbol{\phi}$ to the parameter estimates is now found by ordinary least squares

$$\Delta\boldsymbol{\phi} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \Delta\boldsymbol{y} \,. \tag{14.8}$$

This process is iterated with a new estimate $\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k + \Delta\boldsymbol{\phi}$ until the error $\Delta\boldsymbol{\phi}$ becomes sufficiently small. The Gauss–Newton method has quadratic convergence, which is very fast, provided that there is a good initial estimate $\boldsymbol{\phi}^0$ of the parameters and the nonlinearity is not too severe. An example of a nonlinear model is the kinematic model containing the DH parameters. The nonlinearity is mild because it is due to sines and cosines, so the Gauss–Newton method usually has good convergence properties.

Both for linear and nonlinear estimation, however, rank deficiencies and numerical ill-conditioning may cause problems in inverting $\boldsymbol{A}^\top\boldsymbol{A}$. Rank deficiencies may result from two problems.

1. Inadequate data. The quality of the data for estimating the parameters can be quantified by an observability index, such as the condition number [14.4] of the regressor matrix $\boldsymbol{A}$. A different choice of data to maximize the observability index might result in more robust estimates. Examples are choosing different poses for kinematic calibration, or different trajectories for inertial parameter estimation.
2. Unidentifiable parameters. Perhaps no set of data from an experiment can identify some of the parameters. A procedure has to be found to eliminate or circumvent the unidentifiable parameters. Parameter elimination is usually done by using the singular value decomposition of $\boldsymbol{A}$, while circumvention can be achieved by using a priori values and ridge regression. This does not mean that the parameters are intrinsically identifiable, only that the experimental setup precludes their determination; for example, only one of the 10 inertial parameters of the first link of a robot manipulator can be identified if the base of the manipulator is stationary. A different experimental setup involving accelerating the base of the manipulator and measuring the reaction forces and torques at the base would allow other parameters to be identified [14.5].

Ill-conditioning may result from poor scaling of measurements or parameters.

1. The least-squares estimation minimizes the error between the predicted and measured output vector. Components y_j^l of the output vector $\boldsymbol{y}^l$ may have different units and magnitudes, such as radians versus meters, in pose measurements for kinematic calibration. Moreover, not all components may be measured to the same level of accuracy. Normalizing the output vectors with an appropriately chosen weighting matrix may result in a better conditioned estimate.
2. The parameters may also have different units and different magnitudes. This can cause problems both in terms of convergence criteria and in terms of deciding which parameters to eliminate. Again, a weighting matrix for the parameters may be introduced to improve the conditioning.

These numerical issues are generic to any parameter estimation problem, and are discussed at the end of this chapter. The next sections discuss individual robot models and issues in putting them into parameter estimation form.

14.2 Kinematic Calibration

In general, the relative location between coordinate systems requires six geometric parameters to be calibrated (position plus orientation). If there are mechanical constraints between the relative movement of the coordinate systems, such as connection by a joint, fewer parameters are required. For a rotary joint connecting two links, whose axis is a line-bound vector, four geometric parameters are required. For a prismatic joint, whose axis is a free vector, only two geometric parameters describing orientation are required.

In addition, nongeometric parameters are required to model sensors and mechanical deflection.

- Joint angle sensors may require a gain and offset determination.
- Camera calibration using an undistorted pinhole camera model may require the determination of the focal length and an image sensor offset.
- Joint flexibility from gears leads to angle change due to gravity loading and the manipulator's own weight.
- Base flexibility results from nonrigid attachment of the robot to the environment. Depending on how the manipulator is outstretched, there will be a varying effect on endpoint location.
- Thermal effects and vibration may need to be modeled for very fine positioning control.

This section focuses on determining the geometric parameters and the sensor-based nongeometric parameters.

14.2.1 Serial-Link Robot Manipulators

The modified Denavit–Hartenberg (DH) parameters (Fig. 14.1) serve as the main geometric parameters (Sect. 1.4); the link transformation matrix is

$$
{}^{i-1}\boldsymbol{T}_i = \mathrm{Rot}(\boldsymbol{x}, \alpha_i)\mathrm{Trans}(\boldsymbol{x}, a_i)\mathrm{Trans}(\boldsymbol{z}, d_i)\mathrm{Rot}(\boldsymbol{z}, \theta_i) \,. \tag{14.9}
$$

- For an n-joint manipulator with rotary joint $i = 1, \ldots, n$, whose axis $\boldsymbol{z}_i$ is a line in space, all four parameters a_i, d_i, α_i, and θ_i have to be calibrated.
- For a prismatic joint i, whose axis $\boldsymbol{z}_i$ is a free vector, only two parameters describing its orientation (α_i and θ_i) are required. In principle, the axis $\boldsymbol{z}_i$ can be positioned anywhere in space. This means two DH parameters are arbitrary. One possibility is to intersect $\boldsymbol{z}_i$ with O_{i+1} [14.6, 7], which sets $d_{i+1} = 0$ and $a_{i+1} = 0$. While kinematically correct, this placement is nonintuitive in that it doesn't correspond to the physical location of the prismatic mechanism. It is possible to set a_{i+1} with a_i or θ_i to values that position $\boldsymbol{z}_i$ in the middle of the prismatic joint's mechanical structure.

In the case of nearly parallel neighboring axes, the common normal is poorly defined and the calibration is ill-conditioned. For this case, *Hayati* [14.7] introduced an extra rotational parameter β_i about the $\boldsymbol{y}_{i-1}$ axis (Fig. 14.2).

Let $\boldsymbol{x}'_{i-1}$ lie along a line from O_i to axis $\boldsymbol{z}_{i-1}$, such that $\boldsymbol{x}'_{i-1}$ is perpendicular to $\boldsymbol{z}_i$; the intersection point defines the origin O_{i-1}. Two rotations are required to relate $\boldsymbol{z}_{i-1}$ to $\boldsymbol{z}_i = \boldsymbol{z}'_{i-1}$: a rotation α_i about $\boldsymbol{x}'_{i-1}$ maps $\boldsymbol{z}''_{i-1}$ to $\boldsymbol{z}_i$, and a rotation β_i about $\boldsymbol{y}_{i-1} = \boldsymbol{y}''_{i-1}$ maps $\boldsymbol{z}_{i-1}$ to $\boldsymbol{z}''_{i-1}$. The angle θ_i is now from $\boldsymbol{x}'_{i-1}$ to $\boldsymbol{x}_i$ about $\boldsymbol{z}_i$. The link transformation is

$$
{}^{i-1}\boldsymbol{T}_i = \mathrm{Rot}(\boldsymbol{y}, \beta_i)\mathrm{Rot}(\boldsymbol{x}, \alpha_i)\mathrm{Trans}(\boldsymbol{x}, a_i)\mathrm{Rot}(\boldsymbol{z}, \theta_i) \,. \tag{14.10}
$$

- For a rotary joint, the parameter β_i is calibrated instead of d_i.
- For a prismatic joint, the joint variable d_i has to be retained. As before, position $\boldsymbol{z}_i$ at some convenient location on link i, by specifying two parameters relative to the coordinate system $i+1$. Then proceed with the construction of the Hayati parameters for $d_i = 0$. As d_i changes, axis $\boldsymbol{x}_i$ is displaced along $\boldsymbol{z}_i$ relative to $\boldsymbol{x}'_{i-1}$ (not shown). The link transformation is:

$$
{}^{i-1}\boldsymbol{T}_i = \mathrm{Rot}(\boldsymbol{y}, \beta_i)\mathrm{Rot}(\boldsymbol{x}, \alpha_i)\mathrm{Trans}(\boldsymbol{x}, a_i)\,\mathrm{Trans}(\boldsymbol{z}, d_i)\mathrm{Rot}(\boldsymbol{z}, \theta_i) \,. \tag{14.11}
$$

Although there are five parameters in this transformation, the process of setting the Hayati parameters for $d_i = 0$ is tantamount to setting the value of d_{i-1} to locate O_{i-1}, so there is no net increase in the number of parameters.

The procedures above set the coordinate systems in the intermediate links of a serial manipulator. Coordinate systems also have to be set in the base and end links, but the procedure for doing so depends on the external metrology system and on any physical constraints on the end link pose. The last frame may be n or $n+1$, while the first frame may be 0 or -1 (to keep consecutive numbers); examples appear below. Collect the unknown kinematic parameters into the vectors $\boldsymbol{a}$, $\boldsymbol{d}$, $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, and $\boldsymbol{\beta}$,

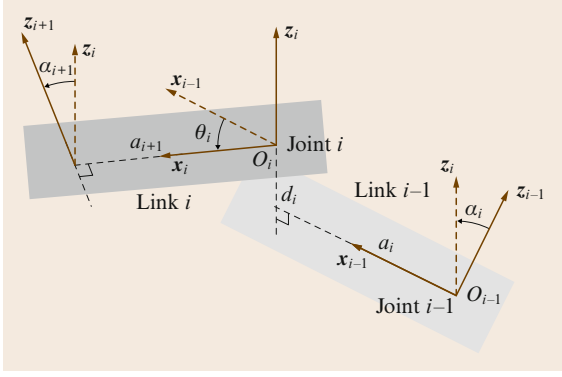

Fig. 14.1 The modified DH parameters

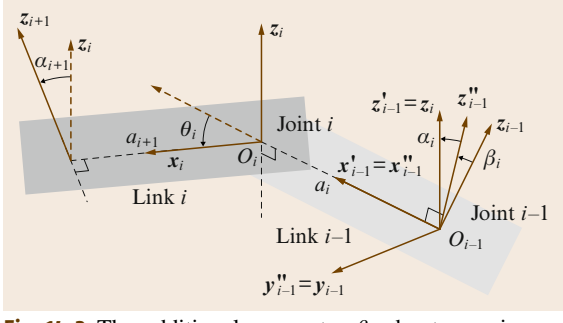

Fig. 14.2 The additional parameter β_i about $\boldsymbol{y}_{i-1}$ is employed for nearly parallel axes

and thence into the parameter vector $\boldsymbol{\phi} = \{\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta}\}$. The parameters $\boldsymbol{\phi}$ predict the position and orientation of the last coordinate system relative to the first, such as ${}^{0}\boldsymbol{T}_{n,c}$.

Not all six components of pose are necessarily used for calibration; the number can vary from one to six. Calibration proceeds by observing the error in the prediction of a certain number of pose components, then employing the nonlinear calibration method (14.7). There are two general methods for observing error.

- *Open-loop calibration* utilizes an external metrology system to measure the pose components. Because the manipulator is not in contact with the environment during this process, the method is termed open loop.
- *Closed-loop calibration* utilizes physical constraints on the end link pose to substitute for measurements. Deviations from the physical constraint represent the prediction error. Because of the contact with the physical constraint, the manipulator forms a closed loop with the ground.

Open-Loop Kinematic Calibration

The calibration literature contains a variety of metrology systems, which can be categorized based on the number of pose components measured [14.8].

1 component: The distance to a single point on the end link can be measured in a variety of ways, such as an instrumented ball bar [14.9], wire potentiometer [14.10], or laser displacement meter [14.11].

2 components: A single theodolite has been employed to provide two orientation measurements [14.12]. A reference length had to be sighted for scaling.

3 components: Laser tracking systems provide accurate three-dimensional (3-D) measurements by reflecting a beam off a retroreflector mounted on the end effector. The beam provides length information, while the gimbal drive for laser steering provides two orientation measurements [14.13]. Since the least precise part of this setup is the angle sensing, another approach is to employ three laser tracking systems and use only the length information. Commercial 3-D stereo-camera motion tracking systems also provide high-accuracy measurements of position.

5 components: *Lau* et al. [14.14] presented a steerable laser interferometer with steerable reflector. With pitch and yaw measurements, the steerable interferometer yields all three components of position, while the steerable reflector yields two components of orientation.

6 components: Full pose can be inferred from the 3-D position of multiple points on the last link measured with a stereo-camera system. A coordinate system fit to the cloud of points yields position plus orientation [14.15].
Vincze et al. [14.16] measured full pose with a single-beam laser tracking system, by fitting the robot with a retroreflector mounted on a universal joint. Position is measured using interferometry, as usual. The novel aspect is orientation measurement, by imaging of the diffraction pattern of the edges of the retroreflector.

Examples are given for calibrating with three pose components and with all six pose components measured.

Point Measurement. The 3-D position of a particular point on the end link can be conveniently located by some form of stereo camera system. The camera system defines a global coordinate system relative to which frame 1 of the robot is located. To provide enough parameters, an intermediate coordinate system has to be introduced; this intermediate coordinate system has index 0, while the camera system is given index −1 to keep consecutive numbers. Two of the eight parameters are arbitrary. Figure 14.3 shows one possibility: z_0 is made

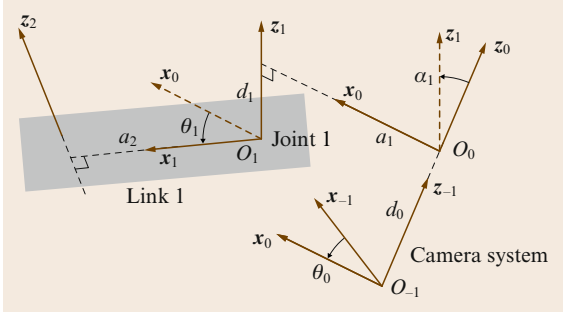

Fig. 14.3 A camera system (index −1) is located relative to the manipulator coordinate system 1 via an intermediate coordinate system 0

parallel ($\alpha_0 = 0$) and coincident ($a_0 = 0$) with z_{-1}. The calibrated parameters are $d_0, \theta_0, a_1, d_1, \alpha_1$, and θ_1. In the case that z_{-1} is nearly parallel to z_0, the measurement coordinate system can be simply redefined to avoid using Hayati parameters; for example, y_{-1} can be redefined as z_{-1}.

In the end link, the origin O_n and axis x_n are unspecified, as are the associated parameters d_n and θ_n. Locating the measured point requires only three parameters, so the addition of a single coordinate system $n+1$ is required to provide an additional parameter. The measured point is defined as the origin O_{n+1} of the new coordinate system, and the normal from z_n that intersects O_{n+1} defines the x_n axis (Fig. 14.4). Three parameters are arbitrary; a simple choice is to make z_{n+1} parallel to z_n ($\alpha_{n+1} = 0$) and x_{n+1} collinear with x_n ($\theta_{n+1} = 0$ and $d_{n+1} = 0$). The calibrated parameters are a_{n+1}, d_n, and θ_n.

From the transformation ${}^{-1}T_{n+1} = {}^{-1}T_0 \ldots {}^{n}T_{n+1}$, the position ${}^{-1}p_{n+1}$ of the measured point relative to the camera frame is extracted. Collect the unknown kinematic parameters into vectors a, d, α, θ, and β, and thence into the parameter vector $\phi = \{a, d, \alpha, \theta, \beta\}$. The nonlinear kinematic model analogous to (14.2) is

$$ {}^{-1}p^l_{n+1} = f(q^l, \phi) , \quad l = 1, \ldots, P , \tag{14.12} $$

where q^l is the vector of joint variables for pose l. To linearize this equation into estimation form (14.6), the associated Jacobians for each parameter type are calculated:

$$ \begin{aligned} \Delta {}^{-1}p^l_{n+1} &= {}^{-1}p^l_{n+1} - {}^{-1}p^l_{n+1,c} = J^l \Delta\phi \\ &= \begin{pmatrix} J^l_a & J^l_d & J^l_\alpha & J^l_\theta & J^l_\beta \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta d \\ \Delta\alpha \\ \Delta\theta \\ \Delta\beta \end{pmatrix} , \end{aligned} \tag{14.13} $$

where a typical column i for each individual parameter Jacobian is derived from the screw parameters as if each parameter represented an active joint (Sect. 1.8.1):

$$ J^l_{a_i} = \begin{cases} {}^{-1}x^l_{i-1} & \text{DH} \\ {}^{-1}x'^l_{i-1} & \text{Hayati} \end{cases} , \tag{14.14} $$

$$ J^l_{d_i} = {}^{-1}z^l_i , \tag{14.15} $$

$$ J^l_{\alpha_i} = \begin{cases} {}^{-1}x^l_{i-1} \times {}^{-1}d^l_{i-1,n+1} & \text{DH} \\ {}^{-1}x'^l_{i-1} \times {}^{-1}d^l_{i-1,n+1} & \text{Hayati} \end{cases} , \tag{14.16} $$

$$ J^l_{\theta_i} = {}^{-1}z^l_i \times {}^{-1}d^l_{i,n+1} , \tag{14.17} $$

$$ J^l_{\beta_i} = {}^{-1}y^l_{i-1} \times {}^{-1}d^l_{i-1,n+1} , \tag{14.18} $$

where ${}^{-1}d_{i,n+1} = {}^{-1}R_i {}^{i}p_{n+1}$ refers the interorigin vector to coordinate system -1. Equations (14.13) are stacked for all the poses for the final estimation form

$$ \Delta {}^{-1}p_{n+1} = J \Delta\phi , \tag{14.19} $$

which corresponds to (14.7) and which is solved for $\Delta\phi$ by least squares and for ϕ by iteration.

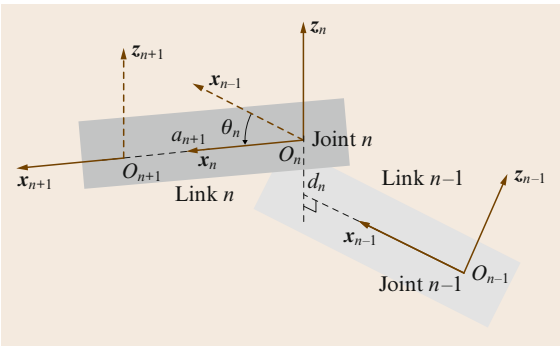

Fig. 14.4 A measured point O_{n+1} is located on the end link by adding a coordinate system $n+1$

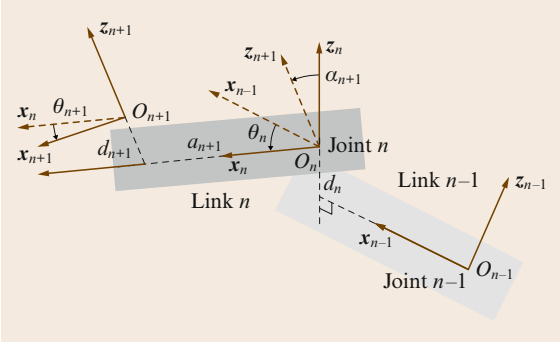

Fig. 14.5 Full pose measurement of coordinate system $n+1$ in the end link

Full Pose Measurement. Suppose that coordinate system $n+1$ in link n is measured (Fig. 14.5). Coordinate system n is completed in the usual way, and the six calibrated parameters to locate coordinate system $n+1$ are d_n, α_n, θ_n, a_{n+1}, d_{n+1}, and θ_{n+1}. If z_{n+1} is nearly parallel to z_n, this axis may simply be permuted with other axes, such as y_{n+1}, to avoid using the Hayati parameters.

In addition to the position equations (14.12), orientation equations of coordinate system $n+1$

$$ {}^{-1}R^l_{n+1} = F(q^l, \phi) , \quad l = 1, \ldots, P \tag{14.20} $$

are extracted from ${}^{-1}\boldsymbol{T}_{n+1}$, where $\boldsymbol{F}$ is a matrix function. Linearization of this equation yields:

$$\begin{aligned}\Delta^{-1}\boldsymbol{R}^l_{n+1} &= {}^{-1}\boldsymbol{R}^l_{n+1} - {}^{-1}\boldsymbol{R}^l_{n+1,c} \\ &= \Delta^{-1}\boldsymbol{\rho}^l_{n+1} \times {}^{-1}\boldsymbol{R}^l_{n+1,c} \\ \boldsymbol{S}(\Delta^{-1}\boldsymbol{\rho}^l_{n+1}) &= \left({}^{-1}\boldsymbol{R}^l_{n+1} - {}^{-1}\boldsymbol{R}^l_{n+1,c}\right)\left({}^{-1}\boldsymbol{R}^l_{n+1,c}\right)^\top ,\end{aligned} \tag{14.21}$$

where $\Delta^{-1}\boldsymbol{\rho}^l_{n+1}$ are differential orthogonal rotations which are the finite-difference counterpart to the angular velocity vector. Continuing with this analogy, a Jacobian $\boldsymbol{J}$ similar to that for spatial velocities (Sect. 1.8) is employed to represent the combined effect of parameter variations $\Delta\boldsymbol{\phi}$ on changes in position $\Delta^{-1}\boldsymbol{p}^l_{n+1}$ and orientation $\Delta^{-1}\boldsymbol{\rho}^l_{n+1}$:

$$\begin{pmatrix}\Delta^{-1}\boldsymbol{p}^l_{n+1} \\ \Delta^{-1}\boldsymbol{\rho}^l_{n+1}\end{pmatrix} = \boldsymbol{J}^l \Delta\boldsymbol{\phi} . \tag{14.22}$$

Compared to (14.13), the Jacobian $\boldsymbol{J}^l$ now has six rows, as do the individual parameter Jacobians:

$$\boldsymbol{J}^l_{a_i} = \begin{cases} \begin{pmatrix} {}^{-1}\boldsymbol{x}^l_{i-1} \\ \boldsymbol{0} \end{pmatrix} & \text{DH} \\ \begin{pmatrix} {}^{-1}\boldsymbol{x'}^l_{i-1} \\ \boldsymbol{0} \end{pmatrix} & \text{Hayati} \end{cases} , \tag{14.23}$$

$$\boldsymbol{J}^l_{d_i} = \begin{pmatrix} {}^{-1}\boldsymbol{z}^l_i \\ \boldsymbol{0} \end{pmatrix} , \tag{14.24}$$

$$\boldsymbol{J}^l_{\alpha_i} = \begin{cases} \begin{pmatrix} {}^{-1}\boldsymbol{x}^l_{i-1} \times {}^{-1}\boldsymbol{d}^l_{i-1,n+1} \\ {}^{-1}\boldsymbol{x}^l_{i-1} \end{pmatrix} & \text{DH} \\ \begin{pmatrix} {}^{-1}\boldsymbol{x'}^l_{i-1} \times {}^{-1}\boldsymbol{d}^l_{i-1,n+1} \\ {}^{-1}\boldsymbol{x'}^l_{i-1} \end{pmatrix} & \text{Hayati} \end{cases} , \tag{14.25}$$

$$\boldsymbol{J}^l_{\theta_i} = \begin{pmatrix} {}^{-1}\boldsymbol{z}^l_i \times {}^{-1}\boldsymbol{d}^l_{i,n+1} \\ {}^{-1}\boldsymbol{z}^l_i \end{pmatrix} , \tag{14.26}$$

$$\boldsymbol{J}^l_{\beta_i} = \begin{pmatrix} {}^{-1}\boldsymbol{y}^l_{i-1} \times {}^{-1}\boldsymbol{d}^l_{i-1,n+1} \\ {}^{-1}\boldsymbol{y}^l_{i-1} \end{pmatrix} . \tag{14.27}$$

As before, $\Delta\boldsymbol{\phi}$ is found by least squares and $\boldsymbol{\phi}$ by iteration.

Closed-Loop Calibration

Physical constraints on end-effector position or orientation can substitute for measurements. The location of a physical constraint defines the reference coordinate system; the *measurement* of endpoint position or orientation is therefore defined as zero. The deviation from the physical constraint due to an incorrect kinematic model is cast as a displacement from the reference coordinates. Analogous to point measurement and full pose measurement, there are closed-loop methods using point constraints and full pose constraints.

Point Constraint. Suppose that the end-effector holds a stylus that makes contact with a fixed point in the environment. The orientation of the stylus can be changed by varying the joint angles, as long as the point of contact is not changed. Before, the measurement system for point measurement defined the reference coordinate system -1 (Fig. 14.3), and there was an end effector coordinate system $n+1$ whose origin O_{n+1} was measured (Fig. 14.4). Now the reference coordinate origin O_0 has index 0 and is collocated with O_{n+1} (Fig. 14.6). An extra coordinate system -1 is not required because a point is located relative to coordinate system 1 with only three parameters: a_1, d_1, and θ_1. The arbitrary choice $\alpha_1 = 0$ is made, i. e., z_0 is made parallel to z_1.

Different poses while maintaining the point contact can be generated manually, or they can be generated automatically if there is a force control capability. In comparison to (14.13), the *measured* position ${}^0\boldsymbol{p}^l_{n+1} = 0$ by definition and the linearized calibration equation is simply:

$$\Delta^{-1}\boldsymbol{p}^l_{n+1} = -{}^{-1}\boldsymbol{p}^l_{n+1,c} = \boldsymbol{J}^l \Delta\boldsymbol{\phi} . \tag{14.28}$$

The set of poses that may be generated is limited relative to open-loop calibration, which may affect identifiability.

Full Pose Constraint. Analogous to full pose measurement (Fig. 14.5), the end link may be fully constrained

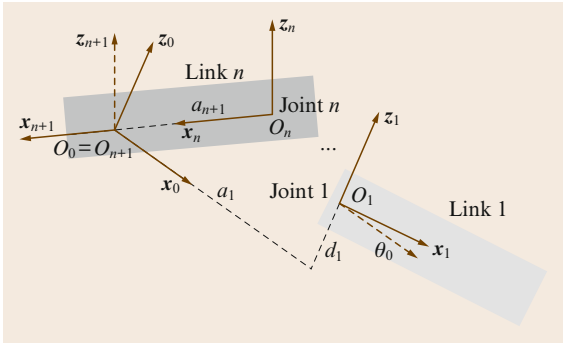

Fig. 14.6 Fixed point contact, setting $O_0 = O_{n+1}$

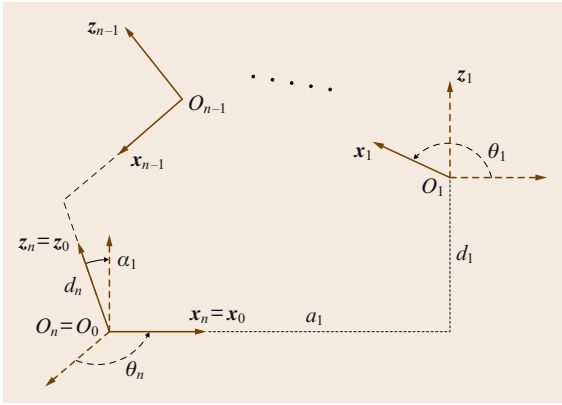

Fig. 14.7 Fully constrained end link, setting $O_0 = O_n$

by rigidly gripping the environment. If the manipulator is redundant (Chap. 11), then poses can be generated through self motion. Generating such poses would require endpoint force/torque sensing or joint torque sensing to be accomplished.

The end link can be considered part of the ground due to rigid attachment, and therefore one fewer coordinate system is required than for the fixed point case. Figure 14.7 shows one way of setting up coordinate systems 0 and n for calibration. Axis z_0 is set equal and coincident with axis z_n. The common normal between z_0 and z_1 sets the origin O_0 and the axis x_0. Coordinate system n is completed by defining $O_n = O_0$ and $x_n = x_0$. Six parameters result for calibration, as they must to relate coordinate system n to coordinate system 0: θ_n, d_n, α_1, a_1, d_1, and θ_1.

Similar to (14.28) after index adjustment, the *measured* position ${}^0\boldsymbol{p}_n^l = 0$ by definition and the linearized position calibration equation is:

$$\Delta {}^0\boldsymbol{p}_n^l = -{}^0\boldsymbol{p}_{n,c}^l \,. \tag{14.29}$$

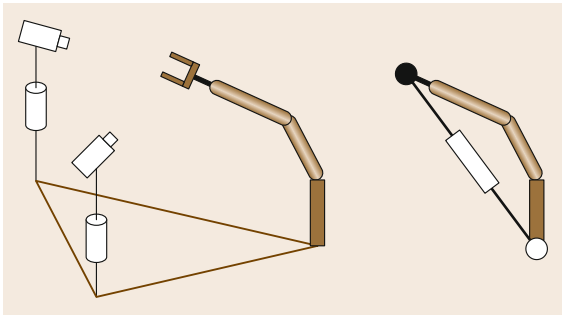

Fig. 14.8 External metrology system modeled as a 6-DOF joint

With regard to the error in orientation (14.21), the *measured* orientation ${}^0\boldsymbol{R}_n^l = \boldsymbol{I}$, the identity matrix, after index adjustment.

$$\boldsymbol{S}\left(\Delta {}^0\boldsymbol{\rho}_n^l\right) = \left(\boldsymbol{I} - {}^0\boldsymbol{R}_{n,c}^l\right)\left({}^0\boldsymbol{R}_{n,c}^l\right)^\top = \left({}^0\boldsymbol{R}_{n,c}^l\right)^\top - \boldsymbol{I} \,. \tag{14.30}$$

The error equation (14.22) is then applied as for full pose measurement, after index adjustment. The set of poses that may be generated is limited relative to open-loop calibration, which may affect identifiability.

14.2.2 Parallel Manipulator Calibration

Parallel manipulators are comprised of multiple closed loops. The methods of the previous section could readily be extended to calibrate parallel manipulators. Rather than treat parallel and serial manipulators differently, and continue to elaborate the calibration equations on a case-by-case basis that includes different loop arrangements and different sensing arrangements, *Hollerbach* and *Wampler* [14.8] presented a unifying methodology termed the *calibration index*, which views all calibration problems as closed-loop calibration problems.

Open-loop calibration is converted to closed-loop calibration by considering the end effector measurement to form a joint. All measurements, whether from joints or metrology systems, are put on an equal footing, as are all unsensed joints, whether from unsensed components of pose, passive environmental constraints, or joints in the chain without sensors. In the case of parallel linkages, sufficient numbers of loop-closure equations are formulated to characterize the kinematics and are combined at each pose. Because the loop-closure equations are implicit functions of all measurements, *Wampler* et al. [14.2] called this calibration method the *implicit loop method*.

Figure 14.8 illustrates this idea for a calibrated stereo camera metrology system that measures the 3-D coordinates of a distinguished point on the end-effector of an uncalibrated robot. On the right this camera system has been replaced by a prismatic leg, which stylistically represents a six-degree-of-freedom (6-DOF) joint that provides equivalent 3-D coordinate measurements. The result is a closed-loop mechanism.

The kinematic loop closure equations $\boldsymbol{f}$ for the ith pose ($i = 1, \ldots, P$) are formed as

$$\boldsymbol{f}^i(\boldsymbol{\phi}) \equiv \boldsymbol{f}(\boldsymbol{x}^i, \boldsymbol{\phi}) = \boldsymbol{0} \,, \tag{14.31}$$

where $\boldsymbol{\phi}$ is a vector of robot parameters to be calibrated, $\boldsymbol{x}^i$ is a vector of joint sensor readings and possibly external sensor readings, and $\boldsymbol{f}^i$ absorbs the sensor readings

$\boldsymbol{x}^i$ and so is given an index. Combining (14.31) for the P poses into a single matrix equation

$$\boldsymbol{f}(\boldsymbol{\phi}) = \left(\boldsymbol{f}^{1\top} \ \ldots \ \boldsymbol{f}^{P\top}\right)^\top = \boldsymbol{0} \,. \tag{14.32}$$

Linearize (14.32) around the nominal values of the parameters

$$\Delta \boldsymbol{f} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\phi}} \Delta \boldsymbol{\phi} = \boldsymbol{A} \Delta \boldsymbol{\phi} \,, \tag{14.33}$$

where $\Delta \boldsymbol{f}$ is the deviation of the computed loop closure equations from zero, $\boldsymbol{J}$ is the identification Jacobian, and $\Delta \boldsymbol{\phi}$ is the correction to be applied to the current parameter estimate. The calibration problem is then solved by minimizing $\Delta \boldsymbol{f}$ via iterative least squares.

The Calibration Index

The basis for kinematic calibration is to compute the error in positioning a manipulator using the current kinematic model. The error may be relative to measurements by an external metrology system (open-loop methods), or relative to a physical constraint such as contact with a plane (closed-loop methods). An external metrology system may measure all six components of pose or fewer components such as a point on the end link (three components). Likewise, a physical constraint can limit between one and six components of pose. Constraints and measurements may be mixed. The number of measurements and constraints determine how many scalar equations per pose are available for calibration.

The calibration index C quantifies the number of equations per pose. While this analysis is rather straightforward for serial linkages, for parallel linkages it can be difficult to infer the number of independent equations per pose

$$C = S - M \,, \tag{14.34}$$

where S is the sensor index and M is the mobility index. The mobility index [14.17] characterizes the degrees of freedom in the calibration setup.

$$M = 6n - \sum_{i=1}^{N_J} n_i^{\mathrm{c}} \,, \tag{14.35}$$

where n is the number of links, N_J is the number of joints, and n_i^{c} is the number of constraints at joint i. n includes any extra links attached to the robot to constrain or measure its motion. N_J includes joints of any additional linkages added for calibration. For a rotational or prismatic joint $n_i^{\mathrm{c}} = 5$, while for a ball or spherical joint $n_i^{\mathrm{c}} = 3$. For an external measurement system for a freely moving end-effector $n_{N_J}^{\mathrm{c}} = 0$, while for rigid attachment of the endpoint $n_{N_J}^{\mathrm{c}} = 6$. While generally correct, there are exceptions to (14.35) due to special or degenerate mechanisms that have to be analyzed on a case-by-case basis (Chap. 12).

The sensor index S is the total number of sensors in the joints:

$$S = \sum_{i=1}^{N_J} S_i \,, \tag{14.36}$$

where S_i is the number of sensed degrees in joint i. Usually $S_i = 1$ for the lower-order pairs typical of actuated joints, while $S_{N_J} = 6$ for full pose measurement of the end-effector *joint* N_J. For an unsensed joint, such as in passive environment kinematics, $S_i = 0$.

If P is the number of poses, then CP is the total number of equations for the calibration procedure. Clearly a larger C means that fewer poses are required, other things being equal. For the single-loop case consisting of a series of sensed lower-order robot joints $(S_i = 1, n_i^{\mathrm{c}} = 5, i = 1, \ldots, N_J - 1)$ with a final joint $(S_{N_J}, n_{N_J}^{\mathrm{c}})$ connecting the end-effector to ground, one has from (14.34–14.35)

$$C = S_{N_J} + n_{N_J}^{\mathrm{c}} \,. \tag{14.37}$$

According to the calibration index, using full endpoint constraints is an equivalent kinematic calibration method to using full pose measurements. There is potentially a problem with the smaller range of poses available with the endpoint constrained, but otherwise the mathematics are the same.

Categorization of Serial Link Calibration Methods

A great variety of calibration methods have been proposed, varying in the manner of pose measurement or endpoint constraints. These are categorized below according to the calibration index C and the values of $n_{N_J}^{\mathrm{c}}$ and S_{N_J}.

- $C = 6$: $n_{N_J}^{\mathrm{c}} = 0$ and $S_{N_J} = 6$ corresponds to full pose measurement. $n_{N_J}^{\mathrm{c}} = 6$ and $S_{N_J} = 0$ corresponds to rigid endpoint attachment.
- $C = 5$: $n_{N_J}^{\mathrm{c}} = 0$ and $S_{N_J} = 5$ corresponds to 5-DOF pose measurement [14.14]. $n_{N_J}^{\mathrm{c}} = 5$ and $S_{N_J} = 0$ corresponds to 5-DOF endpoint constraints, such as manipulation of an unsensed passive hinge joint [14.18].
- $C = 4$: No published method exists for $C = 4$.
- $C = 3$: $n_{N_J}^{\mathrm{c}} = 0$ and $S_{N_J} = 3$ corresponds to 3-DOF pose measurement. $n_{N_J}^{\mathrm{c}} = 3$ and $S_{N_J} = 0$ corresponds to 3-DOF endpoint constraints.

$C = 2$: $n^{c}_{N_J} = 0$ and $S_{N_J} = 2$ corresponds to 2-DOF pose measurements, such as is provided by a single theodolite [14.12]. $n^{c}_{N_J} = 2$ and $S_{N_J} = 0$ corresponds to 2-DOF endpoint constraints. Motion along a line provides two constraints [14.19].

$C = 1$: $n^{c}_{N_J} = 0$ and $S_{N_J} = 1$ corresponds to measurement of just 1-DOF pose, provided by a linear transducer such as an linear variable differential transformer (LVDT) [14.9] or wire potentiometer [14.10]. $n^{c}_{N_J} = 1$ and $S_{N_J} = 0$ corresponds to a plane constraint [14.20].

Categorization of Parallel-Link Calibration Methods

To calibrate parallel robots, a loop-closure equation is written for each loop j:

$$\mathbf{0} = \boldsymbol{f}_j(\boldsymbol{\phi}) \,. \tag{14.38}$$

The equations for all the loops are combined. One problem is to eliminate unsensed degrees of freedom. The calibration index is now applied based on the number of loops.

2 loops: A two-loop mechanism has three arms or legs, attached to a common platform. An example is the RSI Research Ltd. Hand Controller [14.21], which employs three 6-DOF arms with three sensed joints each. The mobility of this mechanism is $M = 6$. As $S = 9$, therefore $C = 3$ and closed-loop calibration is possible.

4 loops: *Nahvi* et al. [14.22] calibrated a spherical shoulder joint, driven redundantly by four prismatic legs (Fig. 14.9). In addition, the platform is constrained to rotate about a spherical joint; with the four legs, four kinematic loops are formed. For this system, $M = 3$ and $S = 4$, so that $C = 1$. Hence self-calibration is possible. Without the extra leg and the sensing it provides, one would have $C = 0$ and calibration would not be possible.

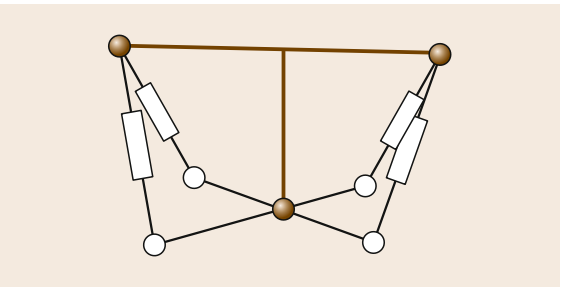

Fig. 14.9 Redundant parallel drive shoulder joint

5 loops: *Wampler* et al. [14.2] calibrated the six-legged Stewart platform ($M = 6$) via a closed-loop procedure. In addition to leg length measurements, all angles on one of the legs were measured ($S = 11$). The reason for the extra sensors is to yield a unique forward kinematics solution, but a side benefit is that, with $C = 5$, closed-loop calibration is possible. For a regular Stewart platform without the instrumented leg, $S = 6$ and hence $C = 0$. External pose measurement is required; for example, with full pose measurement $S = 12$ and $C = 6$; full pose measurement was utilized in [14.23].

14.3 Inertial Parameter Estimation

A rigid body i has 10 inertial parameters: mass m_i, center of mass $\boldsymbol{r}_{0i}$ relative to an origin O_0, and symmetric inertia matrix $\bar{\boldsymbol{I}}_i$ referred here to the origin O_i (Fig. 14.11). The rigid body may be a load held by the end effector, or it can be one of the manipulator's own links. By generating a trajectory and measuring forces or torques in combination with velocity and acceleration, some or all of the inertial parameters can be estimated.

14.3.1 Load Inertial Parameter Estimation

Beginning with load inertial parameter estimation sets the stage for link inertial parameter estimation. It is assumed that there is a wrist-mounted six-axis force/torque sensor, and that suitable filters have been designed to estimate the velocity $\dot{\theta}_i$ and acceleration $\ddot{\theta}_i$ at each joint i. The procedure involves

1. formulating the Newton–Euler equations of the load dynamics to reveal a linear dependence on the inertial parameters, and
2. estimating the parameters using ordinary least squares.

Load Dynamics

Kinematically, the force-torque sensor is part of the last link n, mounted near the joint axis $\boldsymbol{z}_n$ and origin O_n. The

sensor provides readings relative to its own coordinate system, called $n+1$ (Fig. 14.10). The remainder of the last link is attached to the shaft of the force-torque sensor, and so the force-torque sensor is measuring the loads on the last link excluding itself. The acceleration of the force-torque sensor reference frame would have to be calculated based on the amount of offset from O_n, but we will ignore that complication here and assume that the sensor origin O_{n+1} is coincident with O_n.

The center of mass of link n is defined as C_n, located relative to the base origin O_0 by $\boldsymbol{r}_n = C_n - O_0$ and relative to link n's origin O_n by $\boldsymbol{c}_n = C_n - O_n$. All vectors and matrices are expressed in coordinate system n, in which the center of mass location $\boldsymbol{c}_n$ and the inertia matrix $\bar{\boldsymbol{I}}_n$ about origin O_n are constant. The force-torque sensor measures the spatial force $\boldsymbol{f}_n$ at O_n in the coordinate system of the sensor, but the readings are presumed to be converted to the link n coordinates.

From (2.33) in Chap. 2, the Newton–Euler equations referred to O_n are

$$\boldsymbol{f}_n = \boldsymbol{I}_n \boldsymbol{a}_n + \boldsymbol{v}_n \times \boldsymbol{I}_n \boldsymbol{v}_n \,. \tag{14.39}$$

Substituting for the spatial inertia $\boldsymbol{I}_n$, the spatial acceleration $\boldsymbol{a}_n$, and the spatial velocity $\boldsymbol{v}_n$, the first term on the right evaluates to:

$$\begin{aligned}\boldsymbol{I}_n \boldsymbol{a}_n &= \begin{pmatrix} \bar{\boldsymbol{I}}_n & m_n \boldsymbol{S}(\boldsymbol{c}_n) \\ m_n \boldsymbol{S}(\boldsymbol{c}_n)^\top & m_n \boldsymbol{1} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\omega}}_n \\ \dot{\boldsymbol{v}}_n \end{pmatrix} \\ &= \begin{pmatrix} \bar{\boldsymbol{I}}_n \dot{\boldsymbol{\omega}}_n + m_n \boldsymbol{S}(\boldsymbol{c}_n)\left(\ddot{\boldsymbol{d}}_{0n} - \boldsymbol{\omega}_n \times \boldsymbol{v}_n\right) \\ m_n \boldsymbol{S}(\boldsymbol{c}_n)^\top \dot{\boldsymbol{\omega}}_n + m_n \left(\ddot{\boldsymbol{d}}_{0n} - \boldsymbol{\omega}_n \times \boldsymbol{v}_n\right) \end{pmatrix} ,\end{aligned}$$

where $\dot{\boldsymbol{v}}_n = \ddot{\boldsymbol{d}}_{0n} - \boldsymbol{\omega}_n \times \boldsymbol{v}_n$ has been substituted. The second term on the right evaluates to:

$$\begin{aligned}\boldsymbol{v}_n \times \boldsymbol{I}_n \boldsymbol{v}_n &= \begin{pmatrix} \boldsymbol{S}(\boldsymbol{\omega}_n) & \boldsymbol{S}(\boldsymbol{v}_n) \\ \boldsymbol{0} & \boldsymbol{S}(\boldsymbol{\omega}_n) \end{pmatrix} \\ &\quad \times \begin{pmatrix} \bar{\boldsymbol{I}}_n & m_n \boldsymbol{S}(\boldsymbol{c}_n) \\ m_n \boldsymbol{S}(\boldsymbol{c}_n)^\top & m_n \boldsymbol{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega}_n \\ \boldsymbol{v}_n \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{S}(\boldsymbol{\omega}_n) \bar{\boldsymbol{I}}_n \boldsymbol{\omega}_n + m_n \boldsymbol{S}(\boldsymbol{c}_n) \boldsymbol{S}(\boldsymbol{\omega}_n) \boldsymbol{v}_n \\ \boldsymbol{S}(\boldsymbol{\omega}_n) m_n \boldsymbol{S}(\boldsymbol{c}_n)^\top \boldsymbol{\omega}_n + \boldsymbol{S}(\boldsymbol{\omega}_n) m_n \boldsymbol{v}_n \end{pmatrix} .\end{aligned}$$

Combining and simplifying,

$$\boldsymbol{f}_n = \begin{pmatrix} \bar{\boldsymbol{I}}_n \dot{\boldsymbol{\omega}}_n + \boldsymbol{S}(\boldsymbol{\omega}_n) \bar{\boldsymbol{I}}_n \boldsymbol{\omega}_n - \boldsymbol{S}(\ddot{\boldsymbol{d}}_{0n}) m_n \boldsymbol{c}_n \\ m_n \ddot{\boldsymbol{d}}_{0n} + \boldsymbol{S}(\dot{\boldsymbol{\omega}}_n) m_n \boldsymbol{c}_n + \boldsymbol{S}(\boldsymbol{\omega}_n) \boldsymbol{S}(\boldsymbol{\omega}_n) m_n \boldsymbol{c}_n \end{pmatrix} , \tag{14.40}$$

where the mass moment $m_n \boldsymbol{c}_n$ appears as a quantity to be estimated in combination. However, since the mass m_n is separately estimated from the $m_n \ddot{\boldsymbol{d}}_{0n}$ term, the center of mass $\boldsymbol{c}_n$ can be extracted. To account explicitly for gravity $\boldsymbol{g}$, we substitute $\ddot{\boldsymbol{d}}_{0n} - \boldsymbol{g}$ for $\ddot{\boldsymbol{d}}_{0n}$ subsequently.

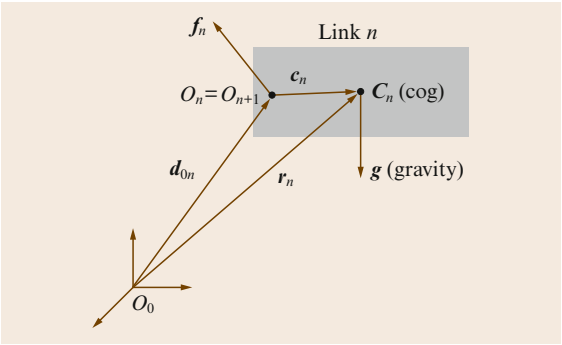

Fig. 14.10 Dynamics of the last link

To formulate an estimation algorithm, the force and torque measured by the wrist sensor must be expressed in terms of the product of known geometric parameters and the unknown inertial parameters. Elements of the inertia matrix are vectorized as $\boldsymbol{l}(\bar{\boldsymbol{I}}_n)$ according to the following notation:

$$\begin{aligned}\bar{\boldsymbol{I}}_n \boldsymbol{\omega}_n &= \begin{pmatrix} \omega_1 & \omega_2 & \omega_3 & 0 & 0 & 0 \\ 0 & \omega_1 & 0 & \omega_2 & \omega_3 & 0 \\ 0 & 0 & \omega_1 & 0 & \omega_2 & \omega_3 \end{pmatrix} \begin{pmatrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{22} \\ I_{23} \\ I_{33} \end{pmatrix} \\ &\equiv \boldsymbol{L}(\boldsymbol{\omega}_n) \boldsymbol{l}\left(\bar{\boldsymbol{I}}_n\right) ,\end{aligned}$$

where $\boldsymbol{L}(\boldsymbol{\omega}_n)$ is a 3×6 matrix of angular velocity elements and

$$\bar{\boldsymbol{I}}_n = \begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{12} & I_{22} & I_{23} \\ I_{13} & I_{23} & I_{33} \end{pmatrix} .$$

Using these expressions, (14.40) can be written as:

$$\begin{aligned}\boldsymbol{f}_n &= \begin{pmatrix} \boldsymbol{0} & -\boldsymbol{S}(\ddot{\boldsymbol{d}}_{0n}) & \boldsymbol{L}(\dot{\boldsymbol{\omega}}_n) + \boldsymbol{S}(\boldsymbol{\omega}_n) \boldsymbol{L}(\boldsymbol{\omega}_n) \\ \ddot{\boldsymbol{d}}_{0n} & \boldsymbol{S}(\dot{\boldsymbol{\omega}}_n) + \boldsymbol{S}(\boldsymbol{\omega}_n) \boldsymbol{S}(\boldsymbol{\omega}_n) & \boldsymbol{0} \end{pmatrix} \\ &\quad \times \begin{pmatrix} m_n \\ m_n \boldsymbol{c}_n \\ \boldsymbol{l}\left(\bar{\boldsymbol{I}}_n\right) \end{pmatrix}\end{aligned}$$

or more compactly,

$$\boldsymbol{f}_n = \boldsymbol{A}_n \boldsymbol{\phi}_n \,, \tag{14.41}$$

where $\boldsymbol{A}_n$ is a 6×10 matrix, and $\boldsymbol{\phi}_n$ is the vector of the 10 unknown inertial parameters which appear linearly.

Estimating the Parameters

The quantities inside the $\boldsymbol{A}_n$ matrix are computed by direct kinematics computation from the measured joint angles, and from the estimated joint velocities and accelerations. The estimation is typically done by bandpass filtering of the joint angle data [14.24]. The elements of the $\boldsymbol{f}_n$ vector are measured directly by the wrist force sensor. For robust estimates in the presence of noise, a larger number of data points are obtained by moving the manipulator through a suitable trajectory. Augment $\boldsymbol{f}_n$ and $\boldsymbol{A}_n$ as:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_n^1 \\ \vdots \\ \boldsymbol{A}_n^P \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} \boldsymbol{f}_n^1 \\ \vdots \\ \boldsymbol{f}_n^P \end{pmatrix}, \tag{14.42}$$

where P is the number of data points. The least-squares estimate for $\boldsymbol{\phi}$ is given by

$$\hat{\boldsymbol{\phi}} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{f} \,. \tag{14.43}$$

Equation (14.43) can also be formulated in a recursive form for online estimation.

For object recognition, it may be desired to derive the inertia about the center of mass, which can be done with the parallel axis theorem. An eigenvalue analysis can diagonalize the inertia matrix to reveal the principal axes and inertia.

14.3.2 Link Inertial Parameter Estimation

By treating each link i of a manipulator as a load, the previous formulation can be extended to the link estimation problem. Unlike load estimation, the only sensing is one component of joint torque. This lack of full force/torque sensing, along with the restricted movement near the base, makes it impossible to find all the inertial parameters of the proximal links. The missing parameters are unimportant, since they have no effect on the control of the arm.

For geared electric drives, joint torque may either be measured by joint torque sensors (Sect. 19.2.4) or estimated from the motor current by using an electric motor model (Sect. 6.3). Most robots do not have joint torque sensors, in which case joint friction needs to be taken into account. Joint friction typically consumes a large fraction of the torque that the motor produces. Coulomb and viscous friction are the most important components of a friction model, although Stribeck friction may need to be modeled for low joint velocities [14.25,26]. A friction model is estimated through a process of moving one joint at a time, and relating motor torque to velocity. Ripple torque, due either to uncompensated inhomogeneities of the magnetic field in the motor [14.27, 28], or to positional dependencies of gear tooth interaction [14.25], may need to be modeled.

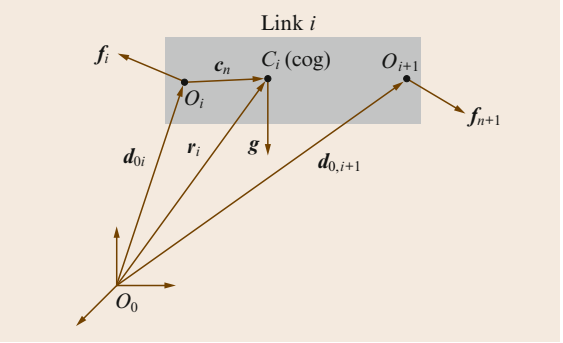

Fig. 14.11 Location of the center of gravity for an intermediate link i, and the constraint forces and torques

Consider a manipulator with n joints (Fig. 14.11). (Only manipulators with revolute joints will be considered, since handling prismatic joints requires only trivial modifications to the algorithm.) Define $\boldsymbol{f}_{ij}$ as the spatial force at joint i due to movement of link j alone. Then $\boldsymbol{f}_{ii}$ is the spatial force at joint i due to movement of its own link, and is the same as (14.41) with i substituted for n in the $\boldsymbol{A}_n$ matrix

$${}^i\boldsymbol{f}_{ii} = {}^i\boldsymbol{A}_i \boldsymbol{\phi}_i \,, \tag{14.44}$$

where $\boldsymbol{\phi}_i$ is the vector of unknown link i inertial parameters. Superscript i has been added to indicate that vectors are expressed in terms of link i coordinates, so that the center of mass ${}^i\boldsymbol{c}_i$ and the inertia matrix ${}^i\bar{\boldsymbol{I}}_i$ are constant.

The total spatial force ${}^i\boldsymbol{f}_i$ at joint i is the sum of the spatial forces ${}^i\boldsymbol{f}_{ij}$ for all links j distal to joint i:

$${}^i\boldsymbol{f}_i = \sum_{j=i}^{n} {}^i\boldsymbol{f}_{ij} \,. \tag{14.45}$$

Each spatial force ${}^i\boldsymbol{f}_{ij}$ at joint i is determined by transmitting the distal spatial force ${}^j\boldsymbol{f}_{jj}$ across intermediate joints. Using the spatial force transform matrix ${}^i\boldsymbol{X}_j^{\mathrm{F}}$,

$${}^i\boldsymbol{f}_{i,i+1} = {}^i\boldsymbol{X}_{i+1}^{\mathrm{F}}\, {}^{i+1}\boldsymbol{f}_{i+1,i+1} = {}^i\boldsymbol{X}_{i+1}^{\mathrm{F}}\, {}^{i+1}\boldsymbol{A}_{i+1}\boldsymbol{\phi}_{i+1} \,. \tag{14.46}$$

For convenience, we note that ${}^i\boldsymbol{X}_i^{\mathrm{F}} = \boldsymbol{1}_{6\times6}$. To obtain the forces and torques at the i-th joint due to the movements

of the j-th link, these matrices can be cascaded:

$$
\begin{aligned}
{}^{i}\boldsymbol{f}_{ij} &= {}^{i}\boldsymbol{X}^{\mathrm{F}}_{i+1}\ {}^{i+1}\boldsymbol{X}^{\mathrm{F}}_{i+2}\ \cdots\ {}^{j-1}\boldsymbol{X}^{\mathrm{F}}_{j}\ {}^{j}\boldsymbol{f}_{jj} \\
&= {}^{i}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j\boldsymbol{\phi}_j\ . \qquad (14.47)
\end{aligned}
$$

An upper-diagonal matrix expression for a serial kinematic chain can be derived from (14.45) and (14.47):

$$
\begin{pmatrix} {}^{1}\boldsymbol{f}_1 \\ {}^{2}\boldsymbol{f}_2 \\ \vdots \\ {}^{n}\boldsymbol{f}_n \end{pmatrix}
= \begin{pmatrix}
{}^{1}\boldsymbol{X}^{\mathrm{F}1}_{1}\boldsymbol{A}_1 & {}^{1}\boldsymbol{X}^{\mathrm{F}2}_{2}\boldsymbol{A}_2 & \cdots & {}^{1}\boldsymbol{X}^{\mathrm{F}n}_{n}\boldsymbol{A}_n \\
\boldsymbol{0} & {}^{2}\boldsymbol{X}^{\mathrm{F}2}_{2}\boldsymbol{A}_2 & \cdots & {}^{2}\boldsymbol{X}^{\mathrm{F}n}_{n}\boldsymbol{A}_n \\
\vdots & \vdots & \ddots & \vdots \\
\boldsymbol{0} & \boldsymbol{0} & \cdots & {}^{n}\boldsymbol{X}^{\mathrm{F}n}_{n}\boldsymbol{A}_n
\end{pmatrix}
\begin{pmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \vdots \\ \boldsymbol{\phi}_n \end{pmatrix} . \qquad (14.48)
$$

This equation is linear in the unknown parameters, but the left-hand side is composed of a full force-torque vector at each joint. Since only the torque τ_i about the joint rotation axis $\boldsymbol{z}_i$ can usually be measured, each spatial force ${}^{i}\boldsymbol{f}_i$ must be projected onto the joint rotation axis, reducing (14.48) to

$$
\boldsymbol{\tau} = \boldsymbol{K}\boldsymbol{\phi}\ , \qquad (14.49)
$$

where

$$
\tau_i = \begin{pmatrix} \boldsymbol{z}_i \\ \boldsymbol{0} \end{pmatrix} \cdot \boldsymbol{f}_i\ ,
$$

$$
\boldsymbol{K}_{ij} = \begin{pmatrix} \boldsymbol{z}_i \\ \boldsymbol{0} \end{pmatrix} \cdot {}^{i}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j\ , \quad \boldsymbol{\phi} = \begin{pmatrix} \boldsymbol{\phi}_1 \\ \cdot \\ \cdot \\ \boldsymbol{\phi}_n \end{pmatrix}
$$

and $\boldsymbol{K}_{ij} = \boldsymbol{0}_{1\times 10}$ if $i > j$. For an n-link manipulator, $\boldsymbol{\tau}$ is an $n\times 1$ vector, $\boldsymbol{\phi}$ is a $10n\times 1$ vector, and $\boldsymbol{K}$ is an $n\times 10n$ matrix.

Equation (14.49) represents the dynamics of the manipulator for one sample point. As with load identification, (14.49) is augmented using P data points:

$$
\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}^1 \\ \vdots \\ \boldsymbol{K}^P \end{pmatrix}\ , \quad \boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{\tau}^1 \\ \vdots \\ \boldsymbol{\tau}^P \end{pmatrix}\ ,
$$

where $\boldsymbol{\tau}$ is now an $nP\times 1$ vector and $\boldsymbol{K}$ is now $nP\times 10n$.

Unfortunately, one cannot apply simple least-squares estimation because $\boldsymbol{K}^{\top}\boldsymbol{K}$ is not invertible due to the loss of rank from restricted degrees of freedom at the proximal links and the lack of full force-torque sensing. Some inertial parameters are completely unidentifiable, while some others can only be identified in linear combinations. The general topic of identifiability of parameters, and how to handle unidentifiable parameters, is discussed next.

An issue with geared electric drives is the rotor inertia. If not known, the rotor inertia can be added to the list of 10 inertial parameters to be identified for a link [14.29]. For large gear ratios, the rotor inertia can dominate the other components of link inertia.

14.3.3 Link Parameter Estimation for More Complex Structures

In this section we present the dynamic identification model for kinematic tree (spanning tree) robots and kinematic closed-loop robots including parallel robots. These models are linear in the inertial parameters and can be represented by a model similar to (14.49), thus these parameters can be identified using the same techniques.

Tree-Structured Robots

For an n-link manipulator with a kinematic tree structure, the dynamic identification model equations (14.48) must be modified to take into account that the inertial parameters of link j have no effect on the dynamics of links which do not belong to the chain from the base to that link. For such structures, the link on which link j is articulated is denoted by $p(j)$ (parent of j – see Chap. 2). It could be a link with any number i, such that $i < j$. Thus the nonzero elements of the column of the matrix representing the coefficients of the inertial parameters of link j in (14.48) would be:

$$
{}^{j}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j,\ {}^{p(j)}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j,\ {}^{p(p(j))}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j, \ldots,\ {}^{b}\boldsymbol{X}^{\mathrm{F}\,j}_{j}\boldsymbol{A}_j\ , \qquad (14.50)
$$

where b is the first link of the chain connecting link 0 to link j, thus $p(b) = 0$.

Consequently, in (14.49) of a tree robot we obtain: $\boldsymbol{K}_{ij} = \boldsymbol{0}_{1\times 10}$ if $i > j$ or if link i does not belong to the branch connecting link 0 to link j. This means that some elements of the upper-right submatrix of $\boldsymbol{K}$ will be zero.

We note that a serial structure is a particular case of a tree structure where $p(j) = j - 1$. Thus any link m such that $m < j$ will belong to the chain from the base to link j.

Closed-Loop Robots

The dynamic model of a closed-loop structure can be obtained by considering first an equivalent spanning tree

by opening one joint at each closed loop, and then using the principle of virtual work such that:

$$\boldsymbol{\tau} = \boldsymbol{G}^{\top} \boldsymbol{K}_{\mathrm{tr}} \boldsymbol{\phi} \,, \quad \boldsymbol{G} = \left(\frac{\partial \boldsymbol{q}_{\mathrm{tr}}}{\partial \boldsymbol{q}_{\mathrm{a}}} \right) \tag{14.51}$$

with $\boldsymbol{q}_{\mathrm{a}}$ an $N \times 1$ vector of N active joint angles (N differs from n, the total number of joints), and $\boldsymbol{q}_{\mathrm{tr}}$ an $n \times 1$ vector of the joints of the spanning tree structure.

Parallel Robots

A parallel robot is a particular case of closed-loop robots (Chap. 12). It is composed of a moving platform, representing the terminal link, that is connected to the base by m parallel legs. The dynamic model of parallel robots can be given by [14.30]

$$\boldsymbol{\tau} = \boldsymbol{J}^{\top} \boldsymbol{A}_p \boldsymbol{\phi}_p + \sum_{i=1}^{m} \left(\frac{\partial \boldsymbol{q}_i}{\partial \boldsymbol{q}_{\mathrm{a}}} \right)^{\top} \boldsymbol{K}_i \boldsymbol{\phi}_i \,, \tag{14.52}$$

where $\boldsymbol{K}_i \boldsymbol{\phi}_i$ represents the dynamic model of leg i with $\boldsymbol{K}_i$ a function of $(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)$, $\boldsymbol{q}_i$ is the vector of joint angles of leg i, $\boldsymbol{A}_p \boldsymbol{\phi}_p$ is the Newton–Euler spatial force of the platform calculated in terms of the Cartesian variables of the mobile platform using (14.41), and $\boldsymbol{J}$ is the Jacobian matrix of the parallel manipulator. We can obtain (14.52) from (14.51) by supposing that the spanning tree structure is obtained by separating the mobile platform from the legs.

Equation (14.52) can be rewritten as

$$\boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{J}^{\top} \boldsymbol{A}_p & (\partial \boldsymbol{q}_1 / \partial \boldsymbol{q}_{\mathrm{a}})^{\top} \boldsymbol{K}_1 & \cdots & (\partial \boldsymbol{q}_m / \partial \boldsymbol{q}_{\mathrm{a}})^{\top} \boldsymbol{K}_m \end{pmatrix} \boldsymbol{\phi}_{\mathrm{par}}$$

$$\boldsymbol{\tau} = \boldsymbol{K}_{\mathrm{par}} \left(\boldsymbol{\omega}_p, \dot{\boldsymbol{v}}_p, \dot{\boldsymbol{\omega}}_p, \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i \right) \boldsymbol{\phi}_{\mathrm{par}} \,, \tag{14.53}$$

where $\boldsymbol{\phi}_{\mathrm{par}}$ is the vector of the inertial parameters of the robot (the legs and the platform).

$$\boldsymbol{\phi}_{\mathrm{par}} = \begin{pmatrix} \boldsymbol{\phi}_p \\ \boldsymbol{\phi}_1 \\ \vdots \\ \boldsymbol{\phi}_m \end{pmatrix} .$$

In the common case where the legs are identical and their inertial parameters are denoted by $\boldsymbol{\phi}_{\mathrm{leg}}$, the identification model can be rewritten by the following equation, which considerably reduces the number of inertial parameters to be identified:

$$\boldsymbol{\tau} = \left(\boldsymbol{J}^{\top} \boldsymbol{A}_p \;\; \sum_{i=1}^{m} (\partial \boldsymbol{q}_i / \partial \boldsymbol{q}_{\mathrm{a}})^{\top} \boldsymbol{K}_i \right) \begin{pmatrix} \boldsymbol{\phi}_p \\ \boldsymbol{\phi}_{\mathrm{leg}} \end{pmatrix} . \tag{14.54}$$

The identification of the orthoglide parallel robot is given in [14.31].

14.4 Identifiability and Numerical Conditioning

That some inertial parameters in (14.49) are unidentifiable does not mean that they are intrinsically identifiable, only that the experimental setup does not allow them to be identified. Limited motion near the base could be fixed by placing the whole manipulator on a six-axis moving platform such as a Stewart–Gough platform. In fact, for mobile manipulators mounted on high-mobility vehicles such as satellites, it may be necessary to know the full inertial model of the manipulator. Additional sensors could be added, for example a six-axis force-torque sensor to the base of the robot [14.5], to identify some additional (but not all) inertial parameters.

A similar situation may arise in kinematic calibration, where for example joint models might be augmented to include gear eccentricities, transmission and coupling coefficients of gears, joint elasticity, link elasticity, and base deflection [14.32]. By instrumenting the robot with additional sensors, for example, by placing joint angle sensors before and after the gears to measure joint deflection, it may be possible to identify additional parameters of interest. Additional sensing with a theodolite was done to measure base deflection in [14.12].

Given that the experimental setup is what it is, there may be no recourse but to deal with unidentifiable parameters, treated in the next section. Another problem is that parameters might in principle be identifiable, but numerical conditioning prevents their accurate determination. Issues such as adequacy of the collected data, and the effect of different units and magnitudes of parameters, are treated subsequently.

14.4.1 Identifiability

There are two main approaches towards handling unidentifiable parameters, depending on whether the goal is a structural model or a prediction model. For a structural model, the goal is to find the minimum

parameter set that provides a meaningful physical description of the system by eliminating parameters until all are identifiable. This is done by careful evaluation of the effect of each parameter of the original model.

For a prediction model, the goal is to match the outputs to the inputs as closely as possible, and is more of a curve-fitting approach. The resulting parameter values are not necessarily physically meaningful or accurate.

Structural Model

It may be possible in the initial model formation to avoid including redundant or unidentifiable parameters. At other times, it is not possible to determine a priori what the minimal parameter set is, because of system complexity or because of numerical difficulties such as measurement error or limitations in the collected data.

A Priori Parameter Elimination. In the initial model formation, the choice of representation can immediately dictate whether the model is redundant or minimal. For kinematic calibration, the minimal parameter set includes four parameters for a rotary joint and two parameters for a prismatic joint. The detailed presentation in Sect. 14.2.1 of how to set up the DH/Hayati parameters for different kinematic and sensing arrangements has the advantage that the parameter set is minimal (excepting joint models). Once done, the application to any manipulator is somewhat formulaic.

Five- and six-parameter joint models have also been proposed, either for the convenience of locating link coordinate systems or for the ease of model formation [14.13,29,33]. With such redundant parameter sets, extra steps must be taken to handle the numerical difficulties caused by the redundancy, such as by reducing the number of parameters in a postprocessing step. When there are complicated joint models that include gearing effects as mentioned above, the resultant large numbers of parameters make the parameter elimination problem more difficult. Determining the minimal set of inertial parameters offers a different sort of complexity, because large numbers of parameters are not identifiable or are identifiable only in linear combinations. Again, model reduction is one approach to handle this problem.

Two approaches towards determining a minimal or base parameter set are

1. numerical identification of the unidentifiable parameters or linear combinations of parameters
2. symbolic determination

Numerical identification involves rank reduction through matrix manipulation of the regressor, either through QR decomposition or singular value decomposition [14.4]. If the kinematic or dynamic model is known exactly, then a simulation using exactly generated *data* will yield a noise-free regressor matrix $\boldsymbol{A}$ in (14.3). For QR decomposition, the regressor matrix is factored as:

$$\boldsymbol{A} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0}_{MP-N_{\mathrm{par}},N_{\mathrm{par}}} \end{pmatrix} , \tag{14.55}$$

where $\boldsymbol{Q}$ is an $M \times MP$ orthogonal matrix, $\boldsymbol{R}$ is an $N_{\mathrm{par}} \times N_{\mathrm{par}}$ upper-triangular matrix, and $\boldsymbol{0}_{MP-N_{\mathrm{par}},N_{\mathrm{par}}}$ is the zero matrix with dimensions $MP - N_{\mathrm{par}} \times N_{\mathrm{par}}$. Theoretically, the unidentifiable parameters ϕ_i are those whose corresponding diagonal elements R_{ii} of matrix $\boldsymbol{R}$ are zero. In practice, $|R_{ii}|$ is considered to be zero if it is less than the numerical zero ζ:

$$\zeta = MP\epsilon \max_i |R_{ii}| , \tag{14.56}$$

where ϵ is the computer precision [14.34]. Other parameters may appear in linear combinations depending on how many nonzero elements there are in row j of $\boldsymbol{R}$. Resolving these linear combinations is arbitrary. One approach is to zero all elements in the linear combination save one. The result is a prediction model rather than a structural model.

Symbolic determination of the base inertial parameters has been proposed in [14.35,36]. Using an energy formulation, the total energy of link j is expressed as $\boldsymbol{h}_j \phi_j$, where $\boldsymbol{h}_j$ is a row vector termed the total energy function whose elements are kinematic expressions of the angular and linear velocity of link j plus gravity. A recursive relation between neighboring links is developed as

$$\boldsymbol{h}_j = \boldsymbol{h}_{j-1}\, {}^{j-1}\boldsymbol{\lambda}_j + \dot{q}_j \boldsymbol{\eta}_j , \tag{14.57}$$

where the elements of the 10×10 matrix ${}^{j-1}\boldsymbol{\lambda}_j$ are functions of the DH parameters defining frame j, and the elements of the 1×10 row vector $\boldsymbol{\eta}_j$ depend on the linear and angular velocities. The details of these expressions can be found in [14.35,36]. Grouping rules are then developed to find the precise linear combinations of parameters.

The minimum inertial parameters for a tree structure can be obtained using a closed-form solution similar to the serial structure case [14.36].

The term $\boldsymbol{K}_{\mathrm{tr}}\boldsymbol{\phi}$ in (14.51) shows that the minimum parameters of the spanning tree structure can be used to calculate the dynamics of the closed structure, giving by this a first reduction and grouping of the inertial parameters. However extra parameters can be eliminated

or grouped when considering the matrix $\boldsymbol{G}$. The case of parallelogram loops can be treated symbolically using closed-form relations [14.36]. For general closed loops, the minimum parameters must be determined using numerical methods such as QR decomposition.

In the case of parallel robots, we deduce from (14.52) that the minimum parameters of the legs can be used to calculate $\boldsymbol{K}_i\boldsymbol{\phi}_i$, giving a first reduction of the inertial parameters. However, some other parameters can be grouped with the parameters of the platform. The minimum parameters of the Gough–Stewart robot are given in [14.37].

Data-Driven Parameter Elimination. The singular value decomposition of the regressor matrix can show which parameters are unidentifiable, weakly identifiable, or identifiable only in linear combination. For N_{par} parameters, P data points, and M-dimensional output measurements at each data point, the regressor matrix $\boldsymbol{A}$ (14.3) or (14.7) can be decomposed as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} , \tag{14.58}$$

where $\boldsymbol{U}$ is an $MP \times MP$ orthogonal matrix, $\boldsymbol{V}$ is an $N_{\text{par}} \times N_{\text{par}}$ orthogonal matrix, and $\boldsymbol{\Sigma}$ is the $MP \times N_{\text{par}}$ matrix of singular values

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{S} \\ \boldsymbol{0}_{MP-N_{\text{par}},N_{\text{par}}} \end{pmatrix} , \tag{14.59}$$

where $\boldsymbol{S} = \text{diag}(\mu_1, \ldots, \mu_r, 0, \ldots, 0)$ is the $N_{\text{par}} \times N_{\text{par}}$ matrix of ordered singular values, with μ_1 the largest and μ_r the smallest nonzero singular value. There may be $N_{\text{par}} - r$ zero singular values $\mu_{r+1} = \cdots = \mu_{N_{\text{par}}} = 0$.

Especially when complex joint models are assumed that include flexibility, backlash, and gear eccentricity, it is not clear that all parameters can be identified. Retaining poorly identifiable parameters will degrade the robustness of the calibration; such parameters are indicated by zero or very small singular values. The expansion of (14.7) in terms of (14.58) is:

$$\Delta\boldsymbol{y} = \sum_{j=1}^{r} \mu_j (\boldsymbol{v}_j^{\top} \Delta\boldsymbol{\phi}) \boldsymbol{u}_j , \tag{14.60}$$

where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are the j-th columns of $\boldsymbol{U}$ and $\boldsymbol{V}$. For zero or small singular values μ_j, the projection $\boldsymbol{v}_j^{\top}\Delta\boldsymbol{\phi}$ of the parameters onto the column vector $\boldsymbol{v}_j$ in general represents a linear combination of parameters. It is also possible that only a single parameter results from the projection.

To proceed, it is first necessary to scale the parameters and the output measurements so that the singular values are comparable. Scaling is discussed in Sect. 14.4.3. Small singular values signal that there are parameters that are poorly identifiable and that should be eliminated. By small singular values, *Schröer* [14.32] suggests the heuristic that the *condition number* of a well-conditioned regressor matrix should be less than 100:

$$\kappa(\boldsymbol{A}) = \frac{\mu_1}{\mu_r} < 100 . \tag{14.61}$$

This heuristic derives from experience of their statistical community. If the condition number is above 100, the singular values are examined, beginning with the smallest one, which may be a zero singular value.

If the condition number is above 100, the linear sums (14.60) corresponding to the smallest singular value μ_r are examined. The elements of column $\boldsymbol{v}_r$ are in one-to-one correspondence with the elements of $\Delta\boldsymbol{\phi}$. If there is an element j of $\boldsymbol{v}_r$ that is much larger than the others, then the parameter ϕ_j corresponding to that column element is a candidate for elimination. This procedure will tend to pinpoint parameters that are totally unidentifiable. Isolating the largest element of $\boldsymbol{v}_r$ is once again only meaningful if the parameters have been previously scaled.

Once the parameter is eliminated, the condition number of the reduced regressor is computed again. The procedure iterates until the condition number of the regressor is below 100.

The previous procedure can be carried out using QR decomposition as in (14.55), by replacing ϵ representing the computer precision in (14.56) by a greater value function of the noise level.

Prediction Model

If there are several largest elements of $\boldsymbol{v}_r$ that are roughly comparable in magnitude, then these parameters may only be identifiable in linear combination. There will be the same number of singular values which are too small. By examining several columns $\boldsymbol{v}_j$ corresponding to the small singular values, these linear combinations can become apparent. The linear combinations can be resolved arbitrarily, i. e., all parameter elements in the linear combination save one can be set to zero. The result of zeroing some parameters is that the model is no longer a structural model, but rather a prediction model.

One may also proceed without first removing parameters. It can be shown by substituting the singular value decomposition that

$$(\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top} = \boldsymbol{V}\left(\boldsymbol{S}^{-1} \;\; \boldsymbol{0}_{N_{\text{par}},MP-N_{\text{par}}}\right)\boldsymbol{U}^{\top} . \tag{14.62}$$

Consequently the solution to (14.8) can be expressed as [14.38]

$$\Delta\boldsymbol{\phi} = \sum_{j=1}^{N_{\text{par}}} \frac{\boldsymbol{u}_j^\top \Delta\boldsymbol{y}}{\mu_j}\boldsymbol{v}_j \,. \tag{14.63}$$

One can see that poorly identifiable parameters corresponding to small singular values μ_j greatly perturb the estimates, because the weighting is $1/\mu_j$. The strategy is to remove their influence. If μ_j is zero or very small relative to the largest singular value μ_1, then set $1/\mu_j = 0$.

Parameters that cannot be identified well are simply ignored in this procedure, which automatically converges to an identifiable parameter set. The resulting parameters can then be used for the model. A disadvantage is that the resulting parameters may not correspond to real model parameters.

Incorporating A Priori Parameter Estimates

Least squares treats parameter values as completely unknown, i. e., they could be anywhere in the range from $-\infty$ to $+\infty$. Yet often a fairly good initial estimate of the parameters is available, for example, from a manufacturer's specifications or in the case of a recalibration. It makes sense to incorporate such an a priori preference into the least-squares optimization [14.39].

Suppose there is a preference for a solution near $\boldsymbol{\phi} = \boldsymbol{\phi}_0$. Express the preference as $\boldsymbol{I}\boldsymbol{\phi} = \boldsymbol{\phi}_0$, where $\boldsymbol{I}$ is the identity matrix, and append it as additional rows to (14.3) that reflect this preference

$$\begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{I} \end{pmatrix} \boldsymbol{\phi} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{\phi}_0 \end{pmatrix} \,. \tag{14.64}$$

To proceed with the solution, we can treat $\boldsymbol{\phi}_0$ as a constant. Redefine the parameter vector as $\tilde{\boldsymbol{\phi}} = \boldsymbol{\phi} - \boldsymbol{\phi}_0$, which we expect to be close to zero. Then

$$\begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{I} \end{pmatrix} \tilde{\boldsymbol{\phi}} = \begin{pmatrix} \tilde{\boldsymbol{y}} \\ \boldsymbol{0} \end{pmatrix} \,, \tag{14.65}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\phi}_0$. We may not know $\boldsymbol{\phi}_0$ perfectly well, so we add a weighting parameter λ to express the confidence in this value

$$\begin{pmatrix} \boldsymbol{A} \\ \lambda\boldsymbol{I} \end{pmatrix} \tilde{\boldsymbol{\phi}} = \begin{pmatrix} \tilde{\boldsymbol{y}} \\ \boldsymbol{0} \end{pmatrix} \,. \tag{14.66}$$

The larger is λ, the more confidence we have in our a priori estimate. The least-squares solution is therefore

$$\begin{aligned} \tilde{\boldsymbol{\phi}} &= \left(\begin{pmatrix} \boldsymbol{A}^\top & \lambda\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{A} \\ \lambda\boldsymbol{I} \end{pmatrix} \right)^{-1} \begin{pmatrix} \boldsymbol{A}^\top & \lambda\boldsymbol{I} \end{pmatrix}^\top \begin{pmatrix} \tilde{\boldsymbol{y}} \\ \boldsymbol{0} \end{pmatrix} \\ &= (\boldsymbol{A}^\top\boldsymbol{A} + \lambda^2\boldsymbol{I})^{-1}\boldsymbol{A}^\top\tilde{\boldsymbol{y}} \,. \end{aligned} \tag{14.67}$$

This solution is called *damped least squares*, where λ is the damping factor. Expanding this solution in terms of the singular value decomposition,

$$\tilde{\boldsymbol{\phi}} = \sum_{j=1}^{N_{\text{par}}} (\boldsymbol{u}_j^\top\tilde{\boldsymbol{y}}) \frac{\mu_j}{\mu_j^2 + \lambda^2}\boldsymbol{v}_j \,. \tag{14.68}$$

Hence a very small μ_j is counteracted by the larger λ value; then the a priori information about parameter values dominates the information from the data, i. e., the data are ignored. Thus for damped least squares, no explicit action on the singular values is required, because the damping factor modifies the singular values. The solution will be perturbed over the normal least-squares solution by the magnitude of the λ choice.

Confidence Measure of Parameter Estimates

After calibration, an estimate $\hat{\boldsymbol{M}}$ of the covariance of the parameter estimates can be derived from the data [14.3]. Assume that the task variables $\Delta\boldsymbol{y}$ have previously been scaled for equal uncertainty, that there is no bias, and that the errors are uncorrelated. Then

$$\hat{\boldsymbol{M}} = \sigma^2(\boldsymbol{A}^\top\boldsymbol{A})^{-1} \,. \tag{14.69}$$

An estimate of the standard deviation σ after running the calibration procedure is obtained from the χ^2 statistic [14.38, 40]:

$$\chi^2 = (\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\hat{\boldsymbol{\phi}})^\top(\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\hat{\boldsymbol{\phi}}) \,. \tag{14.70}$$

An unbiased estimator for σ^2 is $\hat{\sigma}^2 = \chi^2/\nu$, where $\nu = MP - N_{\text{par}}$ is called the statistical *degrees of freedom*; ν subtracts from the total number of measurements MP the number N_{par} of parameters estimated, since some of the measurements went into determining $\boldsymbol{\phi}$.

The estimate $\hat{\boldsymbol{M}}$ can be used as a basis for eliminating parameters, by choosing those with the largest variances.

14.4.2 Observability

The measurements that are taken will influence the accuracy of parameter estimation. In kinematic calibration, the quality of the pose set has been measured by an *observability index*. In inertial parameter estimation, the

quality of the identification trajectory has been termed *persistent excitation* [14.41]. Regardless of whether data is collected statically as in kinematic calibration or dynamically as in inertial parameter estimation, the result is just a bunch of numbers that go into the regressor matrix, and so common terminology is appropriate.

In statistics, optimal experimental design theory has given rise to several data measures termed *alphabet optimalities* [14.42]. Some of the most prominent are the following.

A-optimality	minimizes the trace of $(\boldsymbol{A}^\top \boldsymbol{A})^{-1}$ to obtain regression designs.
D-optimality	maximizes the determinant of $(\boldsymbol{A}^\top \boldsymbol{A})^{-1}$.
E-optimality	maximizes the minimum singular value of $(\boldsymbol{A}^\top \boldsymbol{A})^{-1}$.
G-optimality	minimizes the maximum prediction variance, and does not have a simple expression in terms of singular values.

Although ties to the experimental design literature have not typically been made [14.43], several of the proposed observability indexes for robot calibration have an alphabet-optimality counterpart. A-optimality does not have a counterpart in the robot calibration literature, and conversely a few proposed observability indexes do not have an alphabet-optimality counterpart. In [14.43], it is proved that E-optimality and G-optimality are equivalent for exact design.

Borm and *Menq* [14.44,45] proposed an observability index (here termed O_1 and numbered consecutively below) that maximizes the product of all of the singular values

$$O_1 = \frac{\sqrt[r]{\mu_1 \cdots \mu_r}}{\sqrt{P}} \,. \tag{14.71}$$

This is similar to D-optimality. The rationale is that O_1 represents the volume of a hyperellipsoid in $\Delta \boldsymbol{y}$, defined by (14.7) when $\Delta \boldsymbol{\phi}$ defines a hypersphere, and the singular values represent the lengths of axes. Therefore maximizing O_1 gives the largest hyperellipsoid volume, and hence a good aggregate increase of the singular values. One can also derive O_1 from the well-known relation $\sqrt{\det(\boldsymbol{A}^\top \boldsymbol{A})} = \mu_1 \cdots \mu_r$.

Minimizing the condition number of $\boldsymbol{A}$ as a measure of observability has been proposed in [14.35,46,47]:

$$O_2 = \frac{\mu_1}{\mu_r} \,. \tag{14.72}$$

O_2 measures the eccentricity of the hyperellipsoid rather than its size. The intermediate singular values are not considered to be that pertinent, because minimizing the condition number automatically makes all singular values become more similar in magnitude and makes the hyperellipsoid closer to a hypersphere.

Nahvi et al. [14.22] argue for maximizing the minimum singular value μ_r as an observability measure:

$$O_3 = \mu_r \,. \tag{14.73}$$

This is similar to E-optimality. The rationale is to make the shortest axis as long as possible, regardless of the other axes, that is to say, to improve the worst case. Consider the following standard result [14.4]

$$\mu_r \le \frac{\|\Delta \boldsymbol{y}\|}{\|\Delta \boldsymbol{\phi}\|} \le \mu_1 \tag{14.74}$$

or more particularly

$$\mu_r \|\Delta \boldsymbol{\phi}\| \le \|\Delta \boldsymbol{y}\| \,. \tag{14.75}$$

Then maximizing μ_r ensures that a given size of parameter errors $\|\Delta \boldsymbol{\phi}\|$ has the largest possible influence on the pose errors $\|\Delta \boldsymbol{y}\|$.

Nahvi and *Hollerbach* [14.48] proposed the noise amplification index O_4, which can be viewed as combining the condition number O_2 with the minimum singular value O_3:

$$O_4 = \frac{\sigma_r^2}{\sigma_1} \,. \tag{14.76}$$

The rationale is to measure both the eccentricity of the ellipse through O_2 as well as the size of the ellipse through O_3. The noise amplification index was argued to be the most sensitive to measurement error and modeling error.

Hollerbach and *Lokhorst* [14.21] found experimentally that the condition number and the minimum singular value gave about the same good result: their relative magnitudes were almost proportional to the root-mean-square (RMS) errors of the final parameter errors. The observability index O_1 was not as sensitive and not directly related to parameter errors. [14.43] derived general relations among the observability indexes and alphabet optimalities

$$O_1 \ge \text{A-optimality} \ge O_3 \,. \tag{14.77}$$

They further showed that if $\mu_1 \ge 1$, then $O_3 \ge O_2$, and if $\mu_r \le 1$, then $O_2 \ge O_4$. They also argued that O_3 (D-optimality) is in general the best index, because it minimizes the variance of the parameters and also minimizes the uncertainty of the end-effector pose.

Optimal Experimental Design

An observability index is typically used to decide how many data points to collect. As one begins to add data points, the observability increases, then plateaus. Adding additional data does not then improve the quality of the estimates. For kinematic calibration, data may be added through random selection, or optimal design methods can be employed that can drastically reduce the amount of data required [14.49, 50]. Optimal experimental designs work by measuring the effect of adding or exchanging data points [14.51].

Exciting Trajectories

For inertial parameter estimation, data points are not independent because they result from a trajectory rather than isolated poses. Therefore the issue is what type of trajectory to generate. Industrial robots usually have joint position point-to-point trajectory generators. A continuous and smooth trajectory is calculated by interpolating between these points, assuming zero initial and final velocity and acceleration at each point, and using polynomial interpolators. Excitation trajectories are obtained by minimizing an observability index, using nonlinear optimization techniques to calculate the polynomial coefficients, under the constraints of the joint positions, velocities, and accelerations limits [14.46].

It is possible to facilitate the optimization by proceeding to a sequential excitation procedure. Specific trajectories, which structurally excite a small number of parameters, are easier to optimize. For example, moving one joint at a time with constant velocities excites friction and gravity parameters. In this approach, sequential identification is avoided. However, it is better to collect all the data and to proceed to a globally weighted least-squares estimation [14.24]. This procedure avoids the accumulation of estimation errors and allows the calculation of the confidence intervals (14.69).

Specific trajectories have been proposed like sinusoidal interpolators [14.52], or periodic trajectories obtained from a spectral analysis of the contribution function of the parameters [14.53]. This is a general planning trajectory strategy, which is very important to get accurate experimental identification [14.54].

14.4.3 Scaling

The numerical conditioning of parameter estimation can be improved by scaling both the output measurements (task variable scaling) and the parameters.

Task Variable Scaling

When performing a least-squares analysis on the endpoint pose error, position errors and orientation errors have to be combined (14.22):

$$||\Delta \boldsymbol{y}^i||^2 = ||\Delta^{-1} \boldsymbol{p}^l_{n+1}||^2 + ||\Delta^{-1} \boldsymbol{\rho}^l_{n+1}||^2 \,. \quad (14.78)$$

However, position error and orientation error have different units, and so are not comparable. Furthermore, not all position or orientation components may be measured with equal accuracy.

Ordinary least squares (14.8) weighs all task variables equally. To weigh these variables differently, the general solution is left multiplication of (14.7) by a scaling matrix $\boldsymbol{G}$ [14.39],

$$\begin{aligned} \boldsymbol{G}\Delta \boldsymbol{y} &= \boldsymbol{G}\boldsymbol{A}\Delta \boldsymbol{\phi} \,, \\ \Delta \tilde{\boldsymbol{y}} &\equiv \tilde{\boldsymbol{A}}\Delta \boldsymbol{\phi} \,, \end{aligned} \quad (14.79)$$

where $\Delta \tilde{\boldsymbol{y}} = \boldsymbol{G}\Delta \boldsymbol{y}$ is the scaled output vector and $\tilde{\boldsymbol{A}} = \boldsymbol{G}\boldsymbol{A}$ is the scaled regressor matrix. The weighted least-squares solution is

$$\Delta \boldsymbol{\phi} = (\tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{A}})^{-1} \tilde{\boldsymbol{A}}^\top \Delta \tilde{\boldsymbol{y}} = (\boldsymbol{A}^\top \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{W} \Delta \boldsymbol{y} \,, \quad (14.80)$$

where $\boldsymbol{W} = \boldsymbol{G}^\top \boldsymbol{G}$. Often $\boldsymbol{W}$ is a diagonal matrix.

One approach to scaling position error versus orientation error is to equalize the effect of a parameter error $\Delta\phi_i$ on either position error or orientation error. Curiously, for human-sized robot arms the effect is equal without scaling due to an accident of metric units. If θ is the joint angle resolution, then $s = r\theta$ is the resulting endpoint position resolution. For human-sized arms, $r = 1$ m and hence $s = \theta$. Thus meters and radians are directly comparable after all. Thus using no scaling factors for the pose parameters makes some sense, and may explain why the general disregard for scaling in the robotics community has not had more consequences. If linkages are much smaller (like fingers) or much larger (like excavators) then the situation is different.

A more general way of choosing the weighting matrix $\boldsymbol{W}$ is to use a priori information about acceptable relative errors. Such information might arise from specifications of the measurement apparatus. Suppose that the output variables are subject to independent Gaussian noise, such that σ^y_j is the standard deviation in task variable measurement component Δy^i_j, for $j = 1, \ldots, m$ pose components. Then the individual diagonal weights are $w_{jj} = 1/\sigma^y_j$, and define

$$\begin{aligned} \boldsymbol{R}^i &= \mathrm{diag}\left[\left(\sigma^y_1\right)^2, \ldots, \left(\sigma^y_m\right)^2\right] , \\ \boldsymbol{R} &= \mathrm{diag}(\boldsymbol{R}^1, \ldots, \boldsymbol{R}^P) \,, \end{aligned}$$

where the weighting matrix $\boldsymbol{W} = \boldsymbol{R}^{-1}$, and $\boldsymbol{R}$ is called the *covariance matrix*.

The weighted least-squares estimate is

$$\Delta\boldsymbol{\phi} = (\boldsymbol{A}^\top \boldsymbol{R}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^\top \boldsymbol{R}^{-1}\Delta\boldsymbol{y}\,. \tag{14.81}$$

The resulting scaled output variable $\Delta\tilde{y}_j^l = \Delta y_j^l/\sigma_j^y$ is dimensionless. The larger the uncertainty σ_j^y, the less this variable influences the least-squares solution relative to other variables. The standard deviations σ_j^y are not necessarily the same as endpoint measurement accuracy, because model errors and input noise also contribute to output errors.

The weighted least-squares solution using standard deviations is variously called the Gauss–Markov estimate, the generalized least-squares estimate, or the best linear unbiased estimator (BLUE) [14.3]. It is the minimum covariance estimate (on the parameter error) of all unbiased estimators. A significant point is that the standard deviations of the scaled components of $\Delta\tilde{\boldsymbol{y}}$ are all about the same size, or the covariance $\tilde{\boldsymbol{R}} = \mathrm{cov}(\Delta\tilde{\boldsymbol{y}}) = \boldsymbol{I}$, the identity matrix. Hence the Euclidean norm of the error vector $\Delta\tilde{\boldsymbol{y}}$ is a reasonable measure of its size.

Often we do not know the covariance matrix $\boldsymbol{R}$ that well. An estimate of the standard deviations after running the calibration procedure is obtained from the χ^2 statistic [14.38, 40]:

$$\chi^2 = (\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{\phi})^\top \boldsymbol{R}^{-1}(\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{\phi})\,. \tag{14.82}$$

This equation is the same as the residual error equation (14.79), with substitution for $\boldsymbol{W} = \boldsymbol{R}^{-1}$. χ^2 is nothing more than the weighted residuals after calibration. The expected value of χ^2 is

$$E(\chi^2) \equiv \nu = PK - R\,, \tag{14.83}$$

where E is the expectation operator. That is to say, the unweighted residuals $(\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{\phi})^2$ should, with enough measurements, approximate the true covariance. We may now uniformly scale an initial estimate of $\boldsymbol{R}$ after a preliminary calibration, based on the value of χ^2,

$$\hat{\boldsymbol{R}} = \frac{\chi^2}{\nu}\boldsymbol{R}\,, \tag{14.84}$$

where $\hat{\boldsymbol{R}}$ is the revised estimate of the covariance matrix.

Parameter Scaling

Scaling of parameters is important for proper convergence in nonlinear optimization and for singular value decomposition. If parameters have vastly different magnitudes, then the singular values are not directly comparable. Also, parameter scaling can improve the conditioning of the regressor $\boldsymbol{A}$ and help to avoid invertibility problems.

Whereas left multiplication of $\boldsymbol{A}$ in (14.79) results in task variable scaling, right multiplication of $\boldsymbol{A}$ by a weighting matrix $\boldsymbol{H}$ results in parameter scaling [14.39]:

$$\Delta\boldsymbol{y} = (\boldsymbol{AH})(\boldsymbol{H}^{-1}\Delta\boldsymbol{\phi}) \equiv \bar{\boldsymbol{A}}\Delta\tilde{\boldsymbol{\phi}}\,, \tag{14.85}$$

where the scaled Jacobian and parameters are $\bar{\boldsymbol{A}} = \boldsymbol{AH}$ and $\Delta\tilde{\boldsymbol{\phi}} = \boldsymbol{H}^{-1}\Delta\boldsymbol{\phi}$. The least-squares solution is not changed by parameter scaling, whereas it is by task variable scaling.

The most common approach towards parameter weighting is column scaling, which does not require a priori statistical information. Define a diagonal matrix $\boldsymbol{H} = \mathrm{diag}(h_1, \ldots, h_{N_{\mathrm{par}}})$ with elements

$$h_j = \begin{cases} \|\boldsymbol{c}_j\|^{-1} & \text{if } \|\boldsymbol{c}_j\| \neq 0 \\ 1 & \text{if } \|\boldsymbol{c}_j\| = 0 \end{cases} \tag{14.86}$$

where $\boldsymbol{a}_j$ is the jth column of $\boldsymbol{A}$. Then (14.85) becomes

$$\Delta\boldsymbol{y} = \sum_{j=1}^{N_{\mathrm{par}}} \frac{\boldsymbol{a}_j}{\|\boldsymbol{a}_j\|}\Delta\phi_j\|\boldsymbol{a}_j\|\,. \tag{14.87}$$

Suppose that $\Delta\boldsymbol{y}$ has been previously normalized; then its length is meaningful. Each $\boldsymbol{a}_j/\|\boldsymbol{a}_j\|$ is a unit vector, and hence each scaled parameter $\Delta\phi_j\|\boldsymbol{a}_j\|$ is about the same size and has the same effect on $\Delta\boldsymbol{y}$.

Schröer [14.32] identified a problem with column scaling, namely that poor identifiability of parameters can result in very small Euclidean norms, which result in very large scaling factors. This results in strong amplification of uncertainties of $\boldsymbol{A}$. Instead, Schröer proposed scaling based on the effect on the anticipated error of the robot (as previously discussed under task variable scaling).

In an ideal case, one would have a priori knowledge of the expected value $\boldsymbol{\phi}_0$ of the parameter vector and the standard deviation σ_j^ϕ of each of the parameter vector components. More generally, the parameter distribution would be described by a covariance matrix $\boldsymbol{M}$, but usually this information is not available. Instead, the estimate $\hat{\boldsymbol{M}}$ of the covariance from (14.69) can be used.

If the output measurement covariance $\boldsymbol{R}^{-1}$ and parameter error covariance $\boldsymbol{M}$ are both available, one may define a new least-squares optimization criterion that combines output errors with parameter errors to yield a new χ^2 statistic:

$$\chi^2 = (\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{\phi})^\top \boldsymbol{R}^{-1}(\Delta\boldsymbol{y} - \boldsymbol{A}\Delta\boldsymbol{\phi}) + \Delta\boldsymbol{\phi}^\top \boldsymbol{M}^{-1}\Delta\boldsymbol{\phi}\,. \tag{14.88}$$

Its solution is the minimum-variance estimate, which unlike (14.81) is biased

$$\Delta\boldsymbol{\phi} = (\boldsymbol{A}^\top \boldsymbol{R}^{-1}\boldsymbol{A} + \boldsymbol{M}^{-1})^{-1}\boldsymbol{A}^\top \boldsymbol{R}^{-1}\Delta\boldsymbol{y} \,. \qquad (14.89)$$

The Kalman filter recursively solves the same problem [14.55, 56]; when the state is nonvarying, there is a constant process, and there is no process noise [14.13, 57]. The Gauss–Markov estimate is the limiting case when $\boldsymbol{M}^{-1}$ is zero, i.e., there is no a priori information about the parameters. Again, there is an issue of determining the covariances. As for the Gauss–Markov estimate, the expected value of χ^2 can be used to uniformly scale $\boldsymbol{R}$ and $\boldsymbol{M}$ *post facto* [14.58].

14.5 Conclusions and Further Reading

This chapter has presented methods for calibrating the kinematic parameters and the inertial parameters of robot manipulators. Both methods are instances of parameter estimation using least squares. Inertial parameters appear linearly in the equations of motion, and ordinary least squares can be used. Kinematic parameters appear nonlinearly because of sines and cosines, and so nonlinear estimation via the Gauss–Newton method can be applied.

There are domain-specific issues for setting up the calibration equations. For kinematic calibration, Hayati parameters have to be mixed with Denavit–Hartenberg parameters to handle the case of nearly parallel joint axes. The calibration equations have to take into account how the endpoint is measured or constrained. By examining in a detailed fashion the possible joint sequences, including parallel or prismatic joints, a minimal parameterization is achieved that obviates problems of identifiability.

The calibration index has been presented as categorizing all kinematic calibration methods. This index counts the number of equations generated per pose, by calculating the excess of sensing over mobility. A key is that all calibration methods can be viewed as closed-loop methods, where any endpoint sensing system is considered as a joint. Parallel robots are handled by incorporating multiple closed loops.

For inertial parameter estimation, the link estimation problem has been viewed as an instance of load estimation but with restricted sensing and motion at the joints. The recursive Newton–Euler equations lead to a formulation of the regressor matrix as an upper-diagonal matrix. A minimal parameterization is straightforward for serial and spanning robots.

Numerical methods were presented to handle unidentifiable parameters if a minimal parameterization has not been achieved. These methods hinge on the singular value or QR decomposition of the regressor matrix. The singular values can be examined to decide which parameters are not identifiable and should be eliminated. Alternatively, a small singular value can simply be zeroed to eliminate the effects of poorly identifiable parameters without explicitly eliminating them. The former yields a structural model, while the latter yields a prediction model. A priori estimates of parameters can be taken into account by using damped least squares.

The adequacy of the measurement set for parameter estimation is captured by different proposals for an observability index. This index can be related to the literature on alphabet optimalities in experimental design.

Finally, the scaling of measurements or of parameters is important for a well-conditioned numerical estimation, and is essential in order to compare singular values. When uncertainties in measurements and in parameters are included as weights, the optimal minimum variance estimate can be found, which is related to the Kalman filter. If not known beforehand, these uncertainties can be estimated from the data.

14.5.1 Relation to Other Chapters

Estimation involving least squares and Kalman filtering is discussed in Chap. 4 in a similar context. The problem of estimating properties of the world through sensors is very similar to model identification. Recursive estimation techniques are particularly appropriate when a robot needs to update its world model incrementally. For model identification, employing a recursive formulation is not particularly helpful, as the machinery of recursive updating can obscure numerical issues with the total data.

The singular value decomposition appears in other contexts. Chapters 10 and 11 analyze equal motion capability in different directions by measures similar to the observability indexes: O_1 has a counterpart in the manipulability measure, and O_2 the condition number and O_3 the minimum singular value appear again. By contrast, the concern in calibration is good data in all directions as captured by the singular values. Chapter 11 employs the singular value decomposition to analyze redundant mechanisms. Whereas parameter estimation generally is an overconstrained least-squares problem (many more measurements than parameters), redundant mechanisms are underconstrained (more joint angles than task variables). Instead of signalling identifiability problems, zero singular values indicate the null space of the Jacobian. Damped least squares is used in Chap. 11 to avoid singularities. Just as the true parameters are perturbed by damped least squares in calibration, the trajectory is perturbed in order to get around a numerical conditioning issue.

Sensors involved in positioning a robot can have aspects of their sensor models calibrated as well, such as the gain of a potentiometer. Camera calibration is discussed in Chaps. 23 and 24. Camera models can be determined at the same time as the kinematic models of robot manipulators [14.33, 59], including intrinsic camera parameters such as the pinhole model discussed in Chap. 4 as well as the extrinsic parameters of where the camera is located.

14.5.2 Further Reading

Screw Axis Measurement

An alternative to nonlinear least squares to estimate kinematic parameters is a class of methods which measure the joint axes as lines in space, termed *screw axis measurement* in [14.8]. One approach is *circle point analysis*, which involves moving one joint at a time to generate a circle at a distal measurement point [14.13]. Other approaches measure the Jacobian matrix, which contains as columns the joint screws [14.60]. With knowledge of the joint axes, the kinematic parameters are straightforwardly extracted without nonlinear search. The accuracy of this class of methods may not be as good as the nonlinear least-squares methods.

Total Least Squares

Ordinary least squares assumes that there is only noise in the output measurements, but often there is noise in the inputs as well. Input noise is known to lead to bias errors [14.3]. A framework for handling both input and output noise is total least squares [14.61], also known as orthogonal distance regression [14.62] or errors-in-variables regression [14.63]. Nonlinear total least squares has been applied to robot calibration in [14.2, 64, 65]. In the implicit loop method of [14.2], by treating endpoint measurements equally with joint measurements, no distinction is made between input and output noise.

References

14.1 P.Z. Marmarelis, V.Z. Marmarelis: *Analysis of Physiological Systems* (Plenum, London 1978)

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

14.3 J.P. Norton: *An Introduction to Identification* (Academic, London 1986)

14.4 G.H. Golub, C.F. Van Loan: *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore 1989)

14.5 H. West, E. Papadopoulos, S. Dubowsky, H. Cheah: A method for estimating the mass properties of a manipulator by measuring the reaction moments at its base, Proc. IEEE Int. Conf. Robot. Autom., Scottsdale (IEEE Computer Society Press, Washington 1989) pp. 1510–1516

14.6 R.P. Paul: *Robot Manipulators: Mathematics, Programming, and Control* (MIT Press, Cambridge 1981)

14.7 S.A. Hayati, M. Mirmirani: Improving the absolute positioning accuracy of robot manipulators, J. Robot. Syst. **2**, 397–413 (1985)

14.8 J.M. Hollerbach, C.W. Wampler: The calibration index and taxonomy of kinematic calibration methods, Int. J. Robot. Res. **15**, 573–591 (1996)

14.9 A. Goswami, A. Quaid, M. Peshkin: Identifying robot parameters using partial pose information, IEEE Contr. Syst. **13**, 6–14 (1993)

14.10 M.R. Driels, W.E. Swayze: Automated partial pose measurement system for manipulator calibration experiments, IEEE Trans. Robot. Autom. **10**, 430–440 (1994)

14.11 G.-R. Tang, L.-S. Liu: Robot calibration using a single laser displacement meter, Mechatronics **3**, 503–516 (1993)

14.12 D.E. Whitney, C.A. Lozinski, J.M. Rourke: Industrial robot forward calibration method and results, ASME J. Dyn. Syst. Meas. Contr. **108**, 1–8 (1986)

14.13 B.W. Mooring, Z.S. Roth, M.R. Driels: *Fundamentals of Manipulator Calibration* (Wiley Interscience, New York 1991)

14.14 K. Lau, R. Hocken, L. Haynes: Robot performance measurements using automatic laser tracking

techniques, Robot. Comput.-Integr. Manuf. **2**, 227–236 (1985)
14.15 C.H. An, C.H. Atkeson, J.M. Hollerbach: *Model-Based Control of a Robot Manipulator* (MIT Press, Cambridge 1988)
14.16 M. Vincze, J.P. Prenninger, H. Gander: A laser tracking system to measure position and orientation of robot end effectors under motion, Int. J. Robot. Res. **13**, 305–314 (1994)
14.17 J.M. McCarthy: *Introduction to Theoretical Kinematics* (MIT Press, Cambridge 1990)
14.18 D.J. Bennet, J.M. Hollerbach: Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints, IEEE Trans. Robot. Autom. **7**, 597–606 (1991)
14.19 W.S. Newman, D.W. Osborn: A new method for kinematic parameter calibration via laser line tracking, Proc. IEEE Int. Conf. Robot. Autom., Atlanta, Vol. 2 (IEEE Computer Society Press, Washington 1993) pp. 160–165
14.20 X.-L. Zhong, J.M. Lewis: A new method for autonomous robot calibration, Proc. IEEE Int. Conf. Robot. Autom., Nagoya (IEEE Computer Society Press, Washington 1995) pp. 1790–1795
14.21 J.M. Hollerbach, D.M. Lokhorst: Closed-loop kinematic calibration of the RSI 6-DOF hand controller, IEEE Trans. Robot. Autom. **11**, 352–359 (1995)
14.22 A. Nahvi, J.M. Hollerbach, V. Hayward: Closed-loop kinematic calibration of a parallel-drive shoulder joint, Proc. IEEE Int. Conf. Robot. Autom., San Diego (IEEE Computer Society Press, Washington 1994) pp. 407–412
14.23 O. Masory, J. Wang, H. Zhuang: On the accuracy of a Stewart platform – part II Kinematic calibration and compensation, Proc. IEEE Int. Conf. Robot. Autom., Atlanta (IEEE Computer Society Press, Washington 1994) pp. 725–731
14.24 M. Gautier: Dynamic identification of robots with power model, Proc. IEEE Int. Conf. Robot. Autom. (Albuquerque 1997) pp. 1922–1927
14.25 B. Armstrong-Helouvry: *Control of Machines with Friction* (Kluwer Academic, Boston 1991)
14.26 B. Armstrong-Helouvry, P. Dupont, C. Canudas de Wit: A survey of models, analysis tools and compensation methods for the control of machines with friction, Automatica **30**, 1083–1138 (1994)
14.27 F. Aghili, J.M. Hollerbach, M. Buehler: A modular and high-precision motion control system with an integrated motor, IEEE/ASME Trans. Mechatron. **12**, 317–329 (2007)
14.28 W.S. Newman, J.J. Patel: Experiments in torque control of the Adept One robot, Proc. IEEE Int. Conf. Robot. Autom. (Sacramento 1991) pp. 1867–1872
14.29 W. Khalil, E. Dombre: *Modeling, Identification and Control of Robots* (Taylor Francis, New York 2002)
14.30 W. Khalil, O. Ibrahim: General solution for the dynamic modeling of parallel robots, J. Intell. Robot. Syst. **49**, 19–37 (2007)
14.31 S. Guegan, W. Khalil, P. Lemoine: Identification of the dynamic parameters of the Orthoglide, Proc. IEEE Int. Conf. Robot. Autom. (Taiwan 2003) pp. 3272–3277
14.32 K. Schroer: Theory of kinematic modelling and numerical procedures for robot calibration. In: *Robot Calibration*, ed. by R. Bernhardt, S.L. Albright (Chapman Hall, London 1993) pp. 157–196
14.33 H. Zhuang, Z.S. Roth: *Camera-Aided Robot Calibration* (CRC, Boca Raton 1996)
14.34 J.J. Dongarra, C.B. Mohler, J.R. Bunch, G.W. Stewart: *LINPACK User's Guide* (SIAM, Philadelphia 1979)
14.35 M. Gautier, W. Khalil: Direct calculation of minimum set of inertial parameters of serial robots, IEEE Trans. Robot. Autom. **RA-6**, 368–373 (1990)
14.36 W. Khalil, F. Bennis: Symbolic calculation of the base inertial parameters of closed-loop robots, Int. J. Robot. Res. **14**, 112–128 (1995)
14.37 W. Khalil, S. Guegan: Inverse and direct dynamic modeling of Gough–Stewart robots, Trans. Robot. Autom. **20**, 754–762 (2004)
14.38 W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: *Numerical Recipes in C* (Cambridge Univ. Press, Cambridge 1992)
14.39 C.L. Lawson, R.J. Hanson: *Solving Least Squares Problems* (Prentice Hall, Englewood Cliffs 1974)
14.40 P.R. Bevington, D.K. Robinson: *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill, New York 1992)
14.41 B. Armstrong: On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics, Int. J. Robot. Res. **8**, 28–48 (1989)
14.42 J. Fiefer, J. Wolfowitz: Optimum designs in regression problems, Ann. Math. Stat. **30**, 271–294 (1959)
14.43 Y. Sun, J.M. Hollerbach: Observability index selection for robot calibration, Proc. IEEE Int. Conf. Robot. Autom. (Pasadena 2008), submitted
14.44 J.H. Borm, C.H. Menq: Determination of optimal measurement configurations for robot calibration based on observability measure, Int. J. Robot. Res. **10**, 51–63 (1991)
14.45 C.H. Menq, J.H. Borm, J.Z. Lai: Identification and observability measure of a basis set of error parameters in robot calibration, ASME J. Mechamissions Autom. Des. **111**(4), 513–518 (1989)
14.46 M. Gautier, W. Khalil: Exciting trajectories for inertial parameter identification, Int. J. Robot. Res. **11**, 362–375 (1992)
14.47 M.R. Driels, U.S. Pathre: Significance of observation strategy on the design of robot, J. Robot. Syst. **7**, 197–223 (1990)

14.48 A. Nahvi, J.M. Hollerbach: The noise amplification index for optimal pose selection in robot calibration, Proc. IEEE Int. Conf. Robot. Autom. (1996) pp. 647–654

14.49 D. Daney, B. Madeline, Y. Papegay: Choosing measurement poses for robot calibration with local convergence method and Tabu search, Int. J. Robot. Res. **24**(6), 501–518 (2005)

14.50 Y. Sun, J.M. Hollerbach: Active robot calibration algorithm, Proc. IEEE Int. Conf. Robot. Autom. (Pasadena 2008), submitted

14.51 T.J. Mitchell: An algorithm for the construction of D-Optimal experimental designs, Technometrics **16**(2), 203–210 (1974)

14.52 J. Swevers, C. Ganseman, D.B. Tukel, J. De Schutter, H. Van Brussel: Optimal robot excitation and identification, IEEE Trans. Robot. Autom. **13**, 730–740 (1997)

14.53 P.O. Vandanjon, M. Gautier, P. Desbats: Identification of robots inertial parameters by means of spectrum analysis, Proc. IEEE Int. Conf. Robot. Autom. (Nagoya 1995) pp. 3033–3038

14.54 E. Walter, L. Pronzato: *Identification of Parametric Models from Experimental Data* (Springer, London 1997)

14.55 D.G. Luenberger: *Optimization by Vector Space Methods* (Wiley, New York 1969)

14.56 H.W. Sorenson: Least-squares estimation: from Gauss to Kalman, IEEE Spectrum **7**, 63–68 (1970)

14.57 Z. Roth, B.W. Mooring, B. Ravani: An overview of robot calibration, IEEE J. Robot. Autom. **3**, 377–386 (1987)

14.58 A.E. Bryson Jr., Y.-C. Ho: *Applied Optimal Control* (Hemisphere, Washington 1975)

14.59 D.J. Bennet, J.M. Hollerbach, D. Geiger: Autonomous robot calibration for hand-eye coordination, Int. J. Robot. Res. **10**, 550–559 (1991)

14.60 D.J. Bennet, J.M. Hollerbach, P.D. Henri: Kinematic calibration by direct estimation of the Jacobian matrix, Proc. IEEE Int. Conf. Robot. Autom. (Nice 1992) pp. 351–357

14.61 S. Van Huffel, J. Vandewalle: *The Total Least Squares Problem: Computational Aspects and Analysis* (SIAM, Philadelphia 1991)

14.62 P.T. Boggs, R.H. Byrd, R.B. Schnabel: A stable and efficient algorithm for nonlinear orthogonal distance regression, SIAM J. Sci. Stat. Comput. **8**, 1052–1078 (1987)

14.63 W.A. Fuller: *Measurement Error Models* (Wiley, New York 1987)

14.64 J.-M. Renders, E. Rossignol, M. Becquet, R. Hanus: Kinematic calibration and geometrical parameter identification for robots, IEEE Trans. Robot. Autom. **7**, 721–732 (1991)

14.65 G. Zak, B. Benhabib, R.G. Fenton, I. Saban: Application of the weighted least squares parameter estimation method for robot calibration, J. Mech. Des. **116**, 890–893 (1994)