Chapter 5
Kinetostatics of Serial Robots

5.1 Introduction

Kinetostatics is understood here as the study of the interplay between the *feasible twists* of and the *constraint wrenches* acting on the various rigid bodies of a mechanical system, when the system moves under *static, conservative conditions*. The feasible twists of the various rigid bodies, or links, are those allowed by the constraints imposed by the robot joints. The constraint wrenches are, in turn, the reaction forces and moments exerted on a link by the links to which that link is coupled by means of joints. The subject of this chapter is the kinetostatics of serial robots, with focus on six-axis manipulators. By virtue of the *duality* between the kinematic and the static relations in the mechanics of rigid bodies, as outlined in Sect. 3.7, the derivation of the kinematic relations is discussed in detail, the static relations following from the former.

We derive first the relation between the twist of the robot EE and the set of joint rates, which is given by a linear transformation induced by the robot *Jacobian matrix*. Once the foregoing relation is established for a general six-joint robot, the relation between the static wrench exerted by the environment on the EE and the balancing joint torques is derived by duality. Special robotic architectures are given due attention. Decoupled and planar architectures are treated as special cases of six-joint robots. The fundamental problem of *singularities* arising from a singular robot Jacobian in decoupled manipulators is given due attention as well. Two types of singularities are discussed here for the regional structure of decoupled robots. As a follow-up to the singularity analysis of this structure, its three-dimensional workspace is derived. An algorithm is proposed for the display of this workspace as pertaining to general regional structures whose inverse displacement analysis leads to a quartic polynomial.

Electronic supplementary material The online version of this article (doi: 10.1007/978-3-319-01851-5_5) contains supplementary material, which is available to authorized users.

J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Mechanical Engineering Series 124, DOI 10.1007/978-3-319-01851-5_5,

The chapter closes with a section on kinetostatic performance indices. The purpose of these indices is twofold: They are needed in *robot design* to help the designer best dimension the links of the robot in the early stages of the design process, prior to the *elastostatic* and the *elastodynamic* design stages. These indices are also needed in the control of a given robot to ensure an acceptable kinetostatic performance under feedback control. One third, pragmatic application of these indices is the comparison of various candidate robots when a robotic facility is being planned.

Elastostatic design pertains to the *structural design* of a robot to ensure that the links and the joint mechanical transmissions will be able to withstand the static loads that arise when the robot is in operation. This aspect of design is usually conducted under the assumption that all structural elements operate within the linearly elastic range, and is valid at a specific robot posture. Elastodynamic design considers the inertial load of the structural elements while accounting for link flexibility, which gives rise to mechanical vibration. The main concern here is avoiding resonance under linear dynamical conditions or limit cycles under nonlinear conditions. Both elastostatics and elastodynamics lie beyond the scope of the book, and hence, will not be considered here.

5.2 Velocity Analysis of Serial Manipulators

The relationships between the prescribed twist of the EE, also referred to as the *Cartesian velocity* of the manipulator, and the corresponding joint-rates are derived in this section. First, a serial n-axis manipulator containing only revolute pairs is considered. Then, relations associated with prismatic pairs are introduced, and finally, the joint rates of six-axis manipulators are calculated in terms of the EE twist. Particular attention is given to decoupled manipulators, for which simplified velocity relations are derived.

We consider here the manipulator of Fig. 5.1, in which a joint coordinate θ_i, a joint rate $\dot{\theta}_i$, and a unit vector $\mathbf{e}_i$ are associated with each revolute axis. The X_i, Y_i, Z_i coordinate frame, attached to the $(i-1)$st link, is not shown, but its

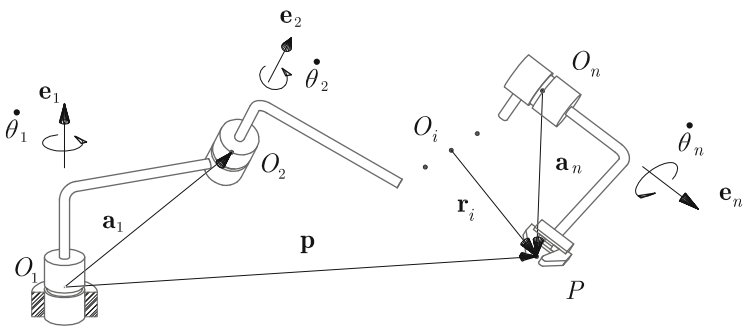


Fig. 5.1 General n-axis manipulator

origin O_i is indicated. If the angular-velocity vector of the ith link is denoted by $\boldsymbol{\omega}_i$, then we have, from Fig. 5.1,

$$\begin{aligned}
\boldsymbol{\omega}_0 &= \mathbf{0} \\
\boldsymbol{\omega}_1 &= \dot{\theta}_1 \mathbf{e}_1 \\
\boldsymbol{\omega}_2 &= \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 \\
&\vdots \\
\boldsymbol{\omega}_n &= \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \cdots + \dot{\theta}_n \mathbf{e}_n
\end{aligned} \tag{5.1}$$

and if the angular velocity of the EE is denoted by $\boldsymbol{\omega}$, then

$$\boldsymbol{\omega} \equiv \boldsymbol{\omega}_n = \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \cdots + \dot{\theta}_n \mathbf{e}_n = \sum_1^n \dot{\theta}_i \mathbf{e}_i$$

Likewise, from Fig. 5.1, one readily derives

$$\mathbf{p} = \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_n \tag{5.2}$$

where $\mathbf{p}$ denotes the position vector of point P of the EE. Moreover, notice that all vectors of the above equation must be expressed in the same frame; otherwise, the addition would not be possible—vector $\mathbf{a}_i$ was defined as expressed in the ith frame in Eq. (4.3c). Upon differentiating both sides of Eq. (5.2), we have

$$\dot{\mathbf{p}} = \dot{\mathbf{a}}_1 + \dot{\mathbf{a}}_2 + \cdots + \dot{\mathbf{a}}_n \tag{5.3}$$

Since vector $\mathbf{a}_i$ is fixed to the ith link,

$$\dot{\mathbf{a}}_i = \boldsymbol{\omega}_i \times \mathbf{a}_i, \quad i = 1, 2, \ldots, n \tag{5.4}$$

Furthermore, substitution of Eqs. (5.1) and (5.4) into Eq. (5.3) yields

$$\begin{aligned}
\dot{\mathbf{p}} = \dot{\theta}_1 \mathbf{e}_1 \times \mathbf{a}_1 + (\dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2) \times \mathbf{a}_2 + \\
\vdots \\
+ (\dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \cdots + \dot{\theta}_n \mathbf{e}_n) \times \mathbf{a}_n
\end{aligned} \tag{5.5}$$

which can be readily rearranged as

$$\begin{aligned}
\dot{\mathbf{p}} = \dot{\theta}_1 \mathbf{e}_1 \times (\mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_n) + \dot{\theta}_2 \mathbf{e}_2 \times (\mathbf{a}_2 + \mathbf{a}_3 + \cdots + \mathbf{a}_n) \\
+ \cdots + \dot{\theta}_n \mathbf{e}_n \times \mathbf{a}_n
\end{aligned}$$

Now vector $\mathbf{r}_i$ is defined as that joining O_i with P, directed from the former to the latter, as depicted in Fig. 5.1, i.e.,

$$\mathbf{r}_i \equiv \mathbf{a}_i + \mathbf{a}_{i+1} + \cdots + \mathbf{a}_n \tag{5.6}$$

and hence, $\dot{\mathbf{p}}$ can be rewritten as

$$\dot{\mathbf{p}} = \sum_{1}^{n} \dot{\theta}_i \mathbf{e}_i \times \mathbf{r}_i$$

Further, let $\mathbf{A}$ and $\mathbf{B}$ denote the $3 \times n$ matrices defined as

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \end{bmatrix} \tag{5.7a}$$

$$\mathbf{B} \equiv \begin{bmatrix} \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \cdots & \mathbf{e}_n \times \mathbf{r}_n \end{bmatrix} \tag{5.7b}$$

the n-dimensional *joint-rate vector* $\dot{\boldsymbol{\theta}}$ being defined, in turn, as

$$\dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \cdots & \dot{\theta}_n \end{bmatrix}^T$$

Thus, $\boldsymbol{\omega}$ and $\dot{\mathbf{p}}$ can be expressed in a more compact form as

$$\boldsymbol{\omega} = \mathbf{A}\dot{\boldsymbol{\theta}}, \quad \dot{\mathbf{p}} = \mathbf{B}\dot{\boldsymbol{\theta}}$$

the twist of the EE being defined, in turn, as

$$\mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{p}} \end{bmatrix} \tag{5.8}$$

The EE twist is thus linearly related to the joint-rate vector $\dot{\boldsymbol{\theta}}$, i.e.,

$$\mathbf{J}\dot{\boldsymbol{\theta}} = \mathbf{t} \tag{5.9}$$

where $\mathbf{J}$ is the *Jacobian matrix*, or *Jacobian*, for brevity, of the manipulator under study, first introduced by Whitney (1972). The Jacobian is defined as the $6 \times n$ matrix shown below:

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \tag{5.10a}$$

or

$$\mathbf{J} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \cdots & \mathbf{e}_n \times \mathbf{r}_n \end{bmatrix} \tag{5.10b}$$

Apparently, an alternative definition of the foregoing Jacobian matrix can be given as

$$\mathbf{J} = \frac{\partial \mathbf{t}}{\partial \dot{\boldsymbol{\theta}}}$$

Moreover, if $\mathbf{j}_i$ denotes the ith column of $\mathbf{J}$, one has

$$\mathbf{j}_i = \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_i \times \mathbf{r}_i \end{bmatrix}$$

It is noteworthy that if the axis of the ith revolute is denoted by $\mathcal{R}_i$, then $\mathbf{j}_i$ is nothing but the Plücker array of that line, with the moment of $\mathcal{R}_i$ being taken with respect to the operation point P of the EE.

On the other hand, if the ith pair is prismatic, then the $(i-1)$st and the ith links have the same angular velocity, for a prismatic pair does not allow any relative rotation. However, vector $\mathbf{a}_i$ joining the origins of the ith and $(i+1)$st frames is no longer of constant magnitude, but undergoes a change of magnitude along the axis of the prismatic pair. This can be appreciated by looking at Fig. 4.6, with b_i playing the role of joint variable. Hence,

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1}, \quad \dot{\mathbf{a}}_i = \boldsymbol{\omega}_{i-1} \times \mathbf{a}_i + \dot{b}_i \mathbf{e}_i$$

One can readily prove, in this case, that

$$\begin{aligned} \boldsymbol{\omega} &= \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \cdots + \dot{\theta}_{i-1} \mathbf{e}_{i-1} + \dot{\theta}_{i+1} \mathbf{e}_{i+1} + \cdots + \dot{\theta}_n \mathbf{e}_n \\ \dot{\mathbf{p}} &= \dot{\theta}_1 \mathbf{e}_1 \times \mathbf{r}_1 + \dot{\theta}_2 \mathbf{e}_2 \times \mathbf{r}_2 + \cdots + \dot{\theta}_{i-1} \mathbf{e}_{i-1} \times \mathbf{r}_{i-1} + \dot{b}_i \mathbf{e}_i \\ &\quad + \dot{\theta}_{i+1} \mathbf{e}_{i+1} \times \mathbf{r}_{i+1} + \cdots + \dot{\theta}_n \mathbf{e}_n \times \mathbf{a}_n \end{aligned}$$

from which it is apparent that the relation between the twist of the EE and the joint-rate vector is formally identical to that appearing in Eq. (5.9) if vector $\dot{\boldsymbol{\theta}}$ is now defined as

$$\dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \cdots & \dot{\theta}_{i-1} & \dot{b}_i & \dot{\theta}_{i+1} & \cdots & \dot{\theta}_n \end{bmatrix}^T$$

the ith column of $\mathbf{J}$ then changing to

$$\mathbf{j}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix} \tag{5.11}$$

Note that the Plücker array of the axis of the ith joint, if prismatic, is that of a line at infinity lying in a plane normal to the unit vector $\mathbf{e}_i$, as defined in Eq. (3.33).

If, in general, $\mathbf{J}_A$ denotes the Jacobian defined for a point A of the EE and $\mathbf{J}_B$ that defined for another point B, then the relation between $\mathbf{J}_A$ and $\mathbf{J}_B$ is

$$\mathbf{J}_B = \mathbf{U}\mathbf{J}_A \tag{5.12a}$$

where the 6×6 matrix $\mathbf{U}$ is defined as

$$\mathbf{U} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{A} - \mathbf{B} & \mathbf{1} \end{bmatrix} \tag{5.12b}$$

while **A** and **B** are now the cross-product matrices of the position vectors **a** and **b** of points A and B, respectively. Moreover, this matrix **U** is identical to the matrix defined under the same name in Eq. (3.29b), and hence, it belongs to the 6×6 unimodular group, i.e., the group of 6×6 matrices whose determinant is unity. Thus,

$$\det(\mathbf{J}_B) = \det(\mathbf{J}_A) \tag{5.13}$$

We have then proven the result below:

Theorem 5.2.1. *The determinant of the Jacobian matrix of a six-axis manipulator is not affected under a change of* operation point *of the EE.*

Note, however, that the Jacobian matrix itself changes under a change of operation point. By analogy with the twist- and the wrench-transfer formulas, Eq. (5.12a) can be called the *Jacobian-transfer formula*.

In particular, for six-axis manipulators, **J** is a 6×6 matrix. Whenever this matrix is nonsingular, Eq. (5.9) can be solved for $\dot{\boldsymbol{\theta}}$, namely,

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1}\mathbf{t} \tag{5.14}$$

Equation (5.14) is only symbolic, for the inverse of the Jacobian matrix need not be computed explicitly. Indeed, in the general case, matrix **J** cannot be inverted symbolically, and hence, $\dot{\boldsymbol{\theta}}$ is computed using a numerical procedure, the most suitable one being the *Gauss-elimination algorithm*, also known as *LU decomposition* (Golub and Van Loan 1989). Gaussian elimination produces the solution by recognizing that a system of linear equations is most easily solved when it is in either upper- or lower-triangular form. To exploit this fact, matrix **J** is decomposed into the *unique* **L** and **U** factors in the form:

$$\mathbf{J} = \mathbf{L}\mathbf{U} \tag{5.15a}$$

where **L** is lower- and **U** is upper-triangular. Moreover, they have the forms

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \tag{5.15b}$$

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \tag{5.15c}$$

where in the particular case at hand, $n = 6$. Thus, the unknown vector of joint rates can now be computed from two triangular systems, namely,

$$\mathbf{L}\mathbf{y} = \mathbf{t}, \quad \mathbf{U}\dot{\boldsymbol{\theta}} = \mathbf{y} \tag{5.16}$$

The above equations are then solved, first for $\mathbf{y}$ and then for $\dot{\boldsymbol{\theta}}$, by application of only forward and backward substitutions, respectively. The LU decomposition of a $n \times n$ matrix requires M'_n multiplications and A'_n additions, whereas the forward substitution needed in solving the lower-triangular system of Eq. (5.16) requires M''_n multiplications and A''_n additions. Moreover, the backward substitution needed in solving the upper-triangular system of Eq. (5.16) requires M'''_n multiplications and A'''_n additions. These figures are (Dahlquist and Björck 1974)

$$M'_n = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}, \qquad A'_n = \frac{n^3}{3} - \frac{n}{3} \tag{5.17a}$$

$$M''_n = \frac{n(n-1)}{2}, \qquad A''_n = \frac{n(n-1)}{2} \tag{5.17b}$$

$$M'''_n = \frac{n(n+1)}{2}, \qquad A'''_n = \frac{n(n-1)}{2} \tag{5.17c}$$

Thus, the solution of a system of n linear equations in n unknowns, using the LU-decomposition method, can be accomplished with M_n multiplications and A_n additions, as given below (Dahlquist and Björck 1974):

$$M_n = \frac{n}{6}(2n^2 + 9n + 1), \quad A_n = \frac{n}{3}(n^2 + 3n - 4) \tag{5.18a}$$

Hence, the velocity resolution of a six-axis manipulator of *arbitrary architecture* requires M_6 multiplications and A_6 additions, i.e.,

$$M_6 = 127, \quad A_6 = 100 \tag{5.18b}$$

5.2.1 Decoupled Manipulators

Decoupled manipulators allow an even simpler velocity resolution. For manipulators with this type of architecture, it is more convenient to deal with the velocity of the center C of the wrist than with that of the operation point P. Thus, one has

$$\mathbf{t}_C = \mathbf{J}\dot{\boldsymbol{\theta}}$$

where $\mathbf{t}_C$ is defined as

$$\mathbf{t}_C = \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{c}} \end{bmatrix}$$

and can be obtained from $\mathbf{t}_P \equiv [\boldsymbol{\omega}^T, \dot{\mathbf{p}}^T]^T$ using the twist-transfer formula given by Eqs. (3.83a) and (3.83b) as

$$\mathbf{t}_C = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{P} - \mathbf{C} & \mathbf{1} \end{bmatrix} \mathbf{t}_P$$

with $\mathbf{C}$ and $\mathbf{P}$ defined as the cross-product matrices of the position vectors $\mathbf{c}$ and $\mathbf{p}$, respectively.

Since C is on the last three joint axes, its velocity is not affected by the motion of the last three joints, and hence, we can write

$$\dot{\mathbf{c}} = \dot{\theta}_1 \mathbf{e}_1 \times \mathbf{r}_1 + \dot{\theta}_2 \mathbf{e}_2 \times \mathbf{r}_2 + \dot{\theta}_3 \mathbf{e}_3 \times \mathbf{r}_3$$

where, in the case of a decoupled manipulator, vector $\mathbf{r}_i$ is defined as that directed from O_i to C. On the other hand, we have

$$\boldsymbol{\omega} = \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \dot{\theta}_3 \mathbf{e}_3 + \dot{\theta}_4 \mathbf{e}_4 + \dot{\theta}_5 \mathbf{e}_5 + \dot{\theta}_6 \mathbf{e}_6$$

and thus, the Jacobian takes on the following simple form

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{O} \end{bmatrix} \tag{5.19}$$

where $\mathbf{O}$ denotes the 3×3 zero matrix, the other 3×3 blocks being given below, *for manipulators with revolute pairs only*, as

$$\mathbf{J}_{11} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \tag{5.20a}$$

$$\mathbf{J}_{12} = \begin{bmatrix} \mathbf{e}_4 & \mathbf{e}_5 & \mathbf{e}_6 \end{bmatrix} \tag{5.20b}$$

$$\mathbf{J}_{21} = \begin{bmatrix} \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \mathbf{e}_3 \times \mathbf{r}_3 \end{bmatrix} \tag{5.20c}$$

Further, vector $\dot{\boldsymbol{\theta}}$ is *partitioned* accordingly:

$$\dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\boldsymbol{\theta}}_a \\ \dot{\boldsymbol{\theta}}_w \end{bmatrix}$$

where

$$\dot{\boldsymbol{\theta}}_a \equiv \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \dot{\boldsymbol{\theta}}_w \equiv \begin{bmatrix} \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix}$$

Henceforth, the three components of $\dot{\boldsymbol{\theta}}_a$ will be referred to as the *arm rates*, whereas those of $\dot{\boldsymbol{\theta}}_w$ will be called the *wrist rates*. Now Eqs. (5.9) can be written, for this particular case, as

$$\mathbf{J}_{11}\dot{\boldsymbol{\theta}}_a + \mathbf{J}_{12}\dot{\boldsymbol{\theta}}_w = \boldsymbol{\omega} \tag{5.21a}$$

$$\mathbf{J}_{21}\dot{\boldsymbol{\theta}}_a = \dot{\mathbf{c}} \tag{5.21b}$$

from which the solution is derived successively from the two systems of three equations and three unknowns that follow:

$$\mathbf{J}_{21}\dot{\boldsymbol{\theta}}_a = \dot{\mathbf{c}} \tag{5.22a}$$

$$\mathbf{J}_{12}\dot{\boldsymbol{\theta}}_w = \boldsymbol{\omega} - \mathbf{J}_{11}\dot{\boldsymbol{\theta}}_a \tag{5.22b}$$

From the general expressions (5.17), then, it is apparent that each of the foregoing systems can be solved with the numbers of operations shown below:

$$M_3 = 23, \quad A_3 = 14$$

Since the computation of the right-hand side of Eq. (5.22b) requires, additionally, nine multiplications and nine additions, the total numbers of operations required to perform one joint-rate resolution of a decoupled manipulator, M_v multiplications and A_v additions, are given by

$$M_v = 55, \quad A_v = 37 \tag{5.23}$$

which are fairly low figures and can be performed in a matter of microseconds using a modern processor.

It is apparent from the foregoing kinematic relations that Eq. (5.22a) should be first solved for $\dot{\boldsymbol{\theta}}_a$; with this value available, Eq. (5.22b) can then be solved for $\dot{\boldsymbol{\theta}}_w$. We thus have, symbolically,

$$\dot{\boldsymbol{\theta}}_a = \mathbf{J}_{21}^{-1}\dot{\mathbf{c}} \tag{5.24}$$

$$\dot{\boldsymbol{\theta}}_w = \mathbf{J}_{12}^{-1}(\boldsymbol{\omega} - \mathbf{J}_{11}\dot{\boldsymbol{\theta}}_a) \tag{5.25}$$

Now, if we recall the concept of reciprocal bases introduced in Sect. 2.7.1, the above inverses can be represented explicitly. Indeed, let

$$\Delta_{21} \equiv \det(\mathbf{J}_{21}) = (\mathbf{e}_1 \times \mathbf{r}_1) \times (\mathbf{e}_2 \times \mathbf{r}_2) \cdot (\mathbf{e}_3 \times \mathbf{r}_3) \tag{5.26}$$

$$\Delta_{12} \equiv \det(\mathbf{J}_{12}) = \mathbf{e}_4 \times \mathbf{e}_5 \cdot \mathbf{e}_6 \tag{5.27}$$

Then,

$$\mathbf{J}_{21}^{-1} = \frac{1}{\Delta_{21}} \begin{bmatrix} [(\mathbf{e}_2 \times \mathbf{r}_2) \times (\mathbf{e}_3 \times \mathbf{r}_3)]^T \\ [(\mathbf{e}_3 \times \mathbf{r}_3) \times (\mathbf{e}_1 \times \mathbf{r}_1)]^T \\ [(\mathbf{e}_1 \times \mathbf{r}_1) \times (\mathbf{e}_2 \times \mathbf{r}_2)]^T \end{bmatrix} \tag{5.28}$$

$$\mathbf{J}_{12}^{-1} = \frac{1}{\Delta_{12}} \begin{bmatrix} (\mathbf{e}_5 \times \mathbf{e}_6)^T \\ (\mathbf{e}_6 \times \mathbf{e}_4)^T \\ (\mathbf{e}_4 \times \mathbf{e}_5)^T \end{bmatrix} \tag{5.29}$$

Therefore,

$$\dot{\boldsymbol{\theta}}_a = \frac{1}{\Delta_{21}} \begin{bmatrix} (\mathbf{e}_2 \times \mathbf{r}_2) \times (\mathbf{e}_3 \times \mathbf{r}_3) \cdot \dot{\mathbf{c}} \\ (\mathbf{e}_3 \times \mathbf{r}_3) \times (\mathbf{e}_1 \times \mathbf{r}_1) \cdot \dot{\mathbf{c}} \\ (\mathbf{e}_1 \times \mathbf{r}_1) \times (\mathbf{e}_2 \times \mathbf{r}_2) \cdot \dot{\mathbf{c}} \end{bmatrix} \tag{5.30a}$$

and, if we let[1]

$$\boldsymbol{\varpi} \equiv \boldsymbol{\omega} - \mathbf{J}_{11} \dot{\boldsymbol{\theta}}_a \tag{5.30b}$$

then

$$\dot{\boldsymbol{\theta}}_w = \frac{1}{\Delta_{12}} \begin{bmatrix} \mathbf{e}_5 \times \mathbf{e}_6 \cdot \boldsymbol{\varpi} \\ \mathbf{e}_6 \times \mathbf{e}_4 \cdot \boldsymbol{\varpi} \\ \mathbf{e}_4 \times \mathbf{e}_5 \cdot \boldsymbol{\varpi} \end{bmatrix} \tag{5.30c}$$

5.3 Jacobian Evaluation

The evaluation of the Jacobian matrix of a manipulator with n revolutes is discussed in this subsection, the presence of a prismatic pair leading to simplifications that will be outlined. Our aim here is to devise algorithms requiring a minimum number of operations, for these calculations are needed either in real-time or in off-line applications when these require Jacobian evaluations at massive numbers of poses, of the order of millions.[2] We assume at the outset that all joint variables producing the desired EE pose are available. We divide this section into two subsections, one for the evaluation of the upper part of the Jacobian matrix and one for the evaluation of its lower part.

[1] $\boldsymbol{\varpi}$ is read *varpi*.

[2] One such application occurs in path planning for machining operations. *Robotmaster* offers a feature that allows the display of maps of the Jacobian condition number, introduced in Sect. 5.8.

5.3.1 Evaluation of Submatrix A

The upper part $\mathbf{A}$ of the Jacobian matrix is composed of the set $\{\,\mathbf{e}_i\,\}_1^n$, our aim here being the calculation of these unit vectors. Note, moreover, that vector $[\,\mathbf{e}_i\,]_1$ is nothing but the last column of $\mathbf{P}_{i-1} \equiv \mathbf{Q}_1 \cdots \mathbf{Q}_{i-1}$, our task then being the calculation of these matrix products. According to the DH notation,

$$[\,\mathbf{e}_i\,]_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

Hence, $[\,\mathbf{e}_1\,]_1$ is available at no cost. However, each of the remaining $[\,\mathbf{e}_i\,]_1$ vectors, for $i = 2, \ldots, n$, is obtained as the last column of matrices $\mathbf{P}_{i-1}$. The *recursive calculation* of these matrices is described below:

$$\begin{aligned} \mathbf{P}_1 &\equiv \mathbf{Q}_1 \\ \mathbf{P}_2 &\equiv \mathbf{P}_1\mathbf{Q}_2 \\ &\vdots \\ \mathbf{P}_n &\equiv \mathbf{P}_{n-1}\mathbf{Q}_n \end{aligned}$$

and hence, a simple algorithm follows:

$$\begin{aligned} &\mathbf{P}_1 \leftarrow \mathbf{Q}_1 \\ &\texttt{For i = 2 to n do} \\ &\quad \mathbf{P}_i \leftarrow \mathbf{P}_{i-1}\mathbf{Q}_i \\ &\texttt{enddo} \end{aligned}$$

Now, since $\mathbf{P}_1$ is identical to $\mathbf{Q}_1$, the first product appearing in the do-loop, $\mathbf{P}_1\mathbf{Q}_2$, is identical to $\mathbf{Q}_1\mathbf{Q}_2$, whose two factors have a special structure. The computation of this product, then, requires special treatment, which warrants further discussion because of its particular features. From the structure of matrices $\mathbf{Q}_i$, as displayed in Eq. (4.1e), we have

$$\mathbf{P}_2 \equiv \begin{bmatrix} \cos\theta_1 & -\lambda_1 \sin\theta_1 & \mu_1 \sin\theta_1 \\ \sin\theta_1 & \lambda_1 \cos\theta_1 & -\mu_1 \cos\theta_1 \\ 0 & \mu_1 & \lambda_1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\lambda_2 \sin\theta_2 & \mu_2 \sin\theta_2 \\ \sin\theta_2 & \lambda_2 \cos\theta_2 & -\mu_2 \cos\theta_2 \\ 0 & \mu_2 & \lambda_2 \end{bmatrix}$$

The foregoing product is calculated now by first computing the products $\lambda_1\lambda_2$, $\lambda_1\mu_2$, $\mu_1\mu_2$, and $\lambda_2\mu_1$, which involve only constant quantities, these terms thus being posture-independent. Thus, in tracking a prescribed Cartesian trajectory, the manipulator posture changes continuously, and hence, its joint variables also change. However, its DH parameters, those defining its architecture, remain constant.

Therefore, the four above products remain constant as well and are computed prior to tracking a trajectory, i.e., *off-line*. In computing these products, we store them as

$$\lambda_{12} \equiv \lambda_1\lambda_2, \quad \mu_{21} \equiv \lambda_1\mu_2, \quad \mu_{12} \equiv \mu_1\mu_2, \quad \lambda_{21} \equiv \lambda_2\mu_1$$

Next, we perform the *on-line* computations. First, let[3]

$$\begin{aligned} \sigma &\leftarrow \lambda_1 \sin\theta_2 \\ \tau &\leftarrow \sin\theta_1 \cos\theta_2 \\ \upsilon &\leftarrow \cos\theta_1 \cos\theta_2 \\ u &\leftarrow \cos\theta_1 \sin\theta_2 + \lambda_1\tau \\ v &\leftarrow \sin\theta_1 \sin\theta_2 - \lambda_1\upsilon \end{aligned}$$

and hence,

$$\mathbf{P}_2 = \begin{bmatrix} \upsilon - \sigma\sin\theta_1 & -\lambda_2 u + \mu_{12}\sin\theta_1 & \mu_2 u + \lambda_{21}\sin\theta_1 \\ \tau + \sigma\cos\theta_1 & -\lambda_2 v - \mu_{12}\cos\theta_1 & \mu_2 v - \lambda_{21}\cos\theta_1 \\ \mu_1\sin\theta_2 & \lambda_{21}\cos\theta_2 + \mu_{21} & -\mu_{12}\cos\theta_2 + \lambda_{12} \end{bmatrix}$$

As the reader can verify, the foregoing calculations consume 20 multiplications and ten additions. Now, we proceed to compute the remaining products in the foregoing do-loop.

Here, notice that the product $\mathbf{P}_{i-1}\mathbf{Q}_i$, for $3 \le i \le n$, can be computed *recursively*, as described below: Let $\mathbf{P}_{i-1}$ and $\mathbf{P}_i$ be given as

$$\mathbf{P}_{i-1} \equiv \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

$$\mathbf{P}_i \equiv \begin{bmatrix} p'_{11} & p'_{12} & p'_{13} \\ p'_{21} & p'_{22} & p'_{23} \\ p'_{31} & p'_{32} & p'_{33} \end{bmatrix}$$

Now matrix $\mathbf{P}_i$ is computed by first defining

$$\begin{aligned} u_i &= p_{11}\sin\theta_i - p_{12}\cos\theta_i \\ v_i &= p_{21}\sin\theta_i - p_{22}\cos\theta_i \\ w_i &= p_{31}\sin\theta_i - p_{32}\cos\theta_i \end{aligned} \tag{5.31a}$$

[3] Although υ and v look similar, they should not be confused with each other, the former being the lowercase Greek letter *upsilon*. As a matter of fact, no confusion should arise, because upsilon is used only once, and does not appear further in the book.

and

$$
\begin{aligned}
p'_{11} &= p_{11}\cos\theta_i + p_{12}\sin\theta_i \\
p'_{12} &= -u_i\lambda_i + p_{13}\mu_i \\
p'_{13} &= u_i\mu_i + p_{13}\lambda_i \\
p'_{21} &= p_{21}\cos\theta_i + p_{22}\sin\theta_i \\
p'_{22} &= -v_i\lambda_i + p_{23}\mu_i \\
p'_{23} &= v_i\mu_i + p_{23}\lambda_i \\
p'_{31} &= p_{31}\cos\theta_i + p_{32}\sin\theta_i \\
p'_{32} &= -w_i\lambda_i + p_{33}\mu_i \\
p'_{33} &= w_i\mu_i + p_{33}\lambda_i
\end{aligned}
\tag{5.31b}
$$

Computing u_i, v_i, and w_i requires six multiplications and three additions, whereas each of the p'_{ij} entries requires two multiplications and one addition. Hence, the computation of each $\mathbf{P}_i$ matrix requires 24 multiplications and 12 additions, the total number of operations required to compute the $n-2$ products $\{\mathbf{P}_i\}_2^{n-1}$ thus being $24(n-2)+20 = 24n-28$ multiplications and $12(n-2)+10 = 12n-14$ additions, for $n \geq 2$. Moreover, $\mathbf{P}_1$, i.e., $\mathbf{Q}_1$, requires four multiplications and no additions, the total number of multiplications M_A and additions A_A required to compute matrix $\mathbf{A}$ thus being

$$
M_A = 24n - 24, \quad A_A = 12n - 14 \tag{5.32}
$$

Before concluding this subsection, a remark is in order: The reader may realize that $\mathbf{P}_n$ is nothing but $\mathbf{Q}$, and hence, the same reader may wonder whether we could not save some operations in the foregoing computations by stopping the above recursive algorithm at $n-1$, rather than at n. This is not a good idea, for the above equality holds if and only if the manipulator is capable of tracking *perfectly* a given trajectory. However, reality is quite different, and errors are always present when tracking. As a matter of fact, the mismatch between $\mathbf{P}_n$ and $\mathbf{Q}$ is very useful in estimating orientation errors, which are then used in a feedback-control scheme to synthesize the corrective signals that are intended to correct those errors.

5.3.2 Evaluation of Submatrix B

The computation of submatrix $\mathbf{B}$ of the Jacobian is studied here. This submatrix comprises the set of vectors $\{\mathbf{e}_i \times \mathbf{r}_i\}_1^n$. We thus proceed first to the computation of vectors $\mathbf{r}_i$, for $i = 1, \ldots, n$, which is most efficiently done using a recursive scheme, similar to that of Horner for polynomial evaluation (Henrici 1964), namely,

$$
\begin{aligned}
&[\,\mathbf{r}_6\,]_6 \leftarrow [\,\mathbf{a}_6\,]_6\\
&\texttt{For i = 5 to 1 do}\\
&\quad [\,\mathbf{r}_i\,]_i \leftarrow [\,\mathbf{a}_i\,]_i + \mathbf{Q}_i[\,\mathbf{r}_{i+1}\,]_{i+1}\\
&\texttt{enddo}
\end{aligned}
$$

In the foregoing algorithm, a simple scheme is introduced to perform the product $\mathbf{Q}_i[\,\mathbf{r}_{i+1}\,]_{i+1}$ economically: if we let $[\,\mathbf{r}_{i+1}\,]_{i+1} = [\,r_1,\, r_2,\, r_3\,]^T$, then

$$
\begin{aligned}
\mathbf{Q}_i[\,\mathbf{r}_{i+1}\,]_{i+1} &= \begin{bmatrix} \cos\theta_i & -\lambda_i \sin\theta_i & \mu_i \sin\theta_i \\ \sin\theta_i & \lambda_i \cos\theta_i & -\mu_i \cos\theta_i \\ 0 & \mu_i & \lambda_i \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}\\
&= \begin{bmatrix} r_1 \cos\theta_i - u \sin\theta_i \\ r_1 \sin\theta_i + u \cos\theta_i \\ r_2\mu_i + r_3\lambda_i \end{bmatrix}
\end{aligned}
\tag{5.33a}
$$

where

$$
u \equiv r_2\lambda_i - r_3\mu_i \tag{5.33b}
$$

Therefore, the product of matrix $\mathbf{Q}_i$ by an arbitrary vector consumes eight multiplications and four additions.

Furthermore, each vector $[\,\mathbf{a}_i\,]_i$, for $i = 1, \ldots, n$, requires two multiplications and no additions, as made apparent from their definitions in Eq. (4.3b). Moreover, from the foregoing evaluation of $\mathbf{Q}_i[\,\mathbf{r}_{i+1}\,]_{i+1}$, it is apparent that each vector $\mathbf{r}_i$, in frame $\mathcal{F}_i$, is computed with ten multiplications and seven additions—two more multiplications are needed to calculate each vector $[\,\mathbf{a}_i\,]_i$ and three more additions are required to add the latter to vector $\mathbf{Q}_i[\,\mathbf{r}_{i+1}\,]_{i+1}$—the whole set of vectors $\{\,\mathbf{r}_i\,\}_1^n$ thus being computed, in $\mathcal{F}_i$-coordinates, with $10(n-1)+2 = 10n-8$ multiplications and $7(n-1)$ additions, where one coordinate transformation, that of $\mathbf{r}_1$, is not counted, since this vector is computed directly in $\mathcal{F}_1$.

Now we turn to the transformation of the components of all the foregoing vectors into $\mathcal{F}_1$-coordinates. First, note that we can proceed now in two ways: in the first, we transform the individual vectors $\mathbf{e}_i$ and $\mathbf{r}_i$ from $\mathcal{F}_i$- into $\mathcal{F}_1$-coordinates and then compute their cross product; in the second, we first perform the cross products and then transform each of these products into $\mathcal{F}_1$-coordinates. It is apparent that the second approach is more efficient, which is why we choose it here.

In order to calculate the products $\mathbf{e}_i \times \mathbf{r}_i$ in $\mathcal{F}_i$-coordinates, we let $[\,\mathbf{r}_i\,]_i = [\,\rho_1,\, \rho_2,\, \rho_3\,]^T$. Moreover, $[\,\mathbf{e}_i\,]_i = [\,0,\, 0,\, 1\,]^T$, and hence,

$$
[\,\mathbf{e}_i \times \mathbf{r}_i\,]_i = \begin{bmatrix} -\rho_2 \\ \rho_1 \\ 0 \end{bmatrix}
$$

which is thus obtained at no cost. Now, the transformation from $\mathcal{F}_i$- into $\mathcal{F}_1$-coordinates is simply

$$[\,\mathbf{e}_i \times \mathbf{r}_i\,]_1 = \mathbf{P}_{i-1}[\,\mathbf{e}_i \times \mathbf{r}_i\,]_i \tag{5.34}$$

In particular, $[\,\mathbf{e}_1 \times \mathbf{r}_1\,]_1$ needs no transformation, for its two factors are given in $\mathcal{F}_1$-coordinates. The $\mathcal{F}_1$-components of the remaining cross products are computed using the general transformation of Eq. (5.34). In the case at hand, this transformation requires, for each i, six multiplications and three additions, for this transformation involves the product of a full 3×3 matrix, $\mathbf{P}_{i-1}$, by a three-dimensional vector, $\mathbf{e}_i \times \mathbf{r}_i$, whose third component vanishes. Thus, the computation of matrix $\mathbf{B}$ requires M_B multiplications and A_B additions, as given below:

$$M_B = 16n - 14, \quad A_B = 10(n-1) \tag{5.35}$$

In total, then, the evaluation of the complete Jacobian requires M_J multiplications and A_J additions, namely,

$$M_J = 40n - 38, \quad A_J = 22n - 24 \tag{5.36}$$

In particular, for a six-revolute manipulator, these figures are 202 multiplications and 108 additions.

Now, if the manipulator contains some prismatic pairs, the foregoing figures diminish correspondingly. Indeed, if the ith joint is prismatic, then the ith column of the Jacobian matrix changes as indicated in Eq. (5.11). Hence, one cross-product calculation is spared, along with the associated coordinate transformation. As a matter of fact, as we saw above, the cross product is computed at no cost in local coordinates, and so each prismatic pair of the manipulator reduces the foregoing numbers of operations by only one coordinate transformation, i.e., by ten multiplications and seven additions.

5.4 Singularity Analysis of Decoupled Manipulators

In performing the computation of the joint rates for a decoupled manipulator, it was assumed that neither $\mathbf{J}_{12}$ nor $\mathbf{J}_{21}$ is singular. If the latter is singular, then none of the joint rates can be evaluated, even if the former is nonsingular. However, if $\mathbf{J}_{21}$ is nonsingular, then Eq. (5.21a) can be solved for the arm rates even if $\mathbf{J}_{12}$ is singular. Each of these sub-Jacobians is analyzed for singularities below.

We will start analyzing $\mathbf{J}_{21}$, whose singularity determines whether any joint-rate resolution is possible at all. First, we note from Eq. (5.20c) that the columns of $\mathbf{J}_{21}$ are the three vectors $\mathbf{e}_1 \times \mathbf{r}_1$, $\mathbf{e}_2 \times \mathbf{r}_2$, and $\mathbf{e}_3 \times \mathbf{r}_3$. Hence, $\mathbf{J}_{21}$ becomes singular if either these three vectors become coplanar or at least one of them vanishes. Furthermore, neither the relative layout of these three vectors nor their magnitudes change if

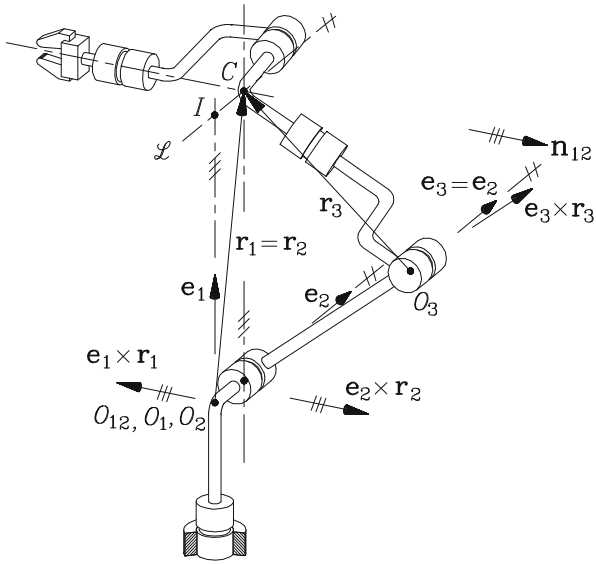
the manipulator undergoes a motion about the first revolute axis while keeping the second and the third revolute axes locked. This means that θ_1 does not affect the singularity of the manipulator, a result that can also be derived from invariance arguments—see Sect. 2.6—and by noticing that singularity is, indeed, an invariant property. Hence, whether a configuration is singular or not is independent of the viewpoint of the observer, a change in θ_1 being nothing but a change of viewpoint. The same argument holds for b_1 in cases where the first joint is prismatic.

The singularity of a three-revolute arm for positioning tasks was analyzed by Burdick (1995), upon recognizing that (a) given three arbitrary lines in space, the three revolute axes in our case, it is always possible to find a set of lines that intersects all three, and (b) the moments of the three lines about any point on the intersecting line are all zero. As a matter of fact, the locus of those lines is a quadric ruled surface, namely, a one-sheet hyperboloid—see Exercise 3.3. Therefore, if the endpoint of the third moving link lies in this quadric, the manipulator is in a singular posture, and velocities of C along the intersecting line cannot be produced. This means that the manipulator has lost, to a *first order*, one degree of freedom. Here we emphasize that this loss is meaningful only at a first order because, in fact, a motion along that intersecting line may still be possible, provided that the full nonlinear relations of Eq. (4.16) are considered. If such a motion is at all possible, however, then it is so only in one direction, as we shall see in Case 2 below. Motions in the opposite direction are not feasible because of the rigidity of the links.

We will illustrate the foregoing concepts as pertaining to the most common types of industrial manipulators, i.e., those of the orthogonal type. In these cases, two consecutive axes either intersect at right angles or are parallel; most of the time, the first two axes intersect at right angles and the last two are parallel. Below we study each of these cases separately.

Case 1: Two consecutive axes intersect and C lies in their plane. Here, the ruled hyperboloid containing the lines that intersect all three axes degenerates into a plane, namely, that of the two intersecting axes. For conciseness, let us assume that the first two axes intersect, but the derivations are the same if the intersecting axes are the last two. Moreover, let O_{12} be the intersection of the first two axes, Π_{12} being the plane of these axes and $\mathbf{n}_{12}$ its normal. If we recall the notation adopted in Sect. 5.2, we have now that the vector directed from O_{12} to C can be regarded as both $\mathbf{r}_1$ and $\mathbf{r}_2$. Furthermore, $\mathbf{e}_1 \times \mathbf{r}_1$ and $\mathbf{e}_2 \times \mathbf{r}_2$ $(= \mathbf{e}_2 \times \mathbf{r}_1)$ are both parallel to $\mathbf{n}_{12}$. Hence, the first two axes can only produce velocities of C in the direction of $\mathbf{n}_{12}$. As a consequence, velocities of C in Π_{12} and perpendicular to $\mathbf{e}_3 \times \mathbf{r}_3$ cannot be produced in the presence of this singularity. The set of unfeasible velocities, then, lies in a line normal to $\mathbf{n}_{12}$ and $\mathbf{e}_3 \times \mathbf{r}_3$, whose direction is the geometric representation of the null space of $\mathbf{J}_{21}^T$. Likewise, the manipulator can withstand forces applied at C in the direction of the same line purely by reaction wrenches, i.e., without any motor torques. The last issue falls into the realm of manipulator statics, upon which we will elaborate in Sect. 5.6.

We illustrate this singularity, termed here *shoulder singularity*, in a manipulator with the architecture of Fig. 4.3, as postured in Fig. 5.2. In this figure, the line



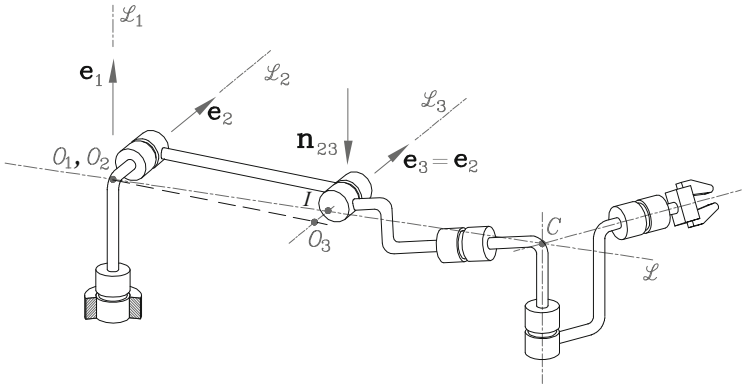


Fig. 5.2 Shoulder singularity of the Puma robot

Fig. 5.3 Elbow singularity of the Puma robot

intersecting all three arm axes is not as obvious and needs further explanation. This line is indicated by $\mathcal{L}$ in that figure, and is parallel to the second and third axes. It is apparent that this line intersects the first axis at right angles at a point I. Now, if we take into account that all parallel lines intersect at infinity, then it becomes apparent that $\mathcal{L}$ intersects the axes of the second and third revolutes as well, and hence, $\mathcal{L}$ intersects all three axes.

Case 2: Two consecutive axes are parallel and C lies in their plane, as shown in Fig. 5.3. For conciseness, again, we assume that axes 2 and 3 are parallel, a rather common case in industrial manipulators, but the derivations below are the same if the parallel axes are the first two. We now let Π_{23} be the plane of the last

two axes and $\mathbf{n}_{23}$ its normal. Furthermore, $\mathbf{e}_3 = \mathbf{e}_2$ and $\mathbf{r}_2 = \mathbf{r}_1$; moreover, the intersection of $\overline{O_2C}$ with $\mathcal{L}_3$ is point I of the same figure, while $\overline{O_3I} = d$; then, $\mathbf{r}_3 = d\,\mathbf{e}_3 + \alpha\mathbf{r}_2$ and $\mathbf{e}_3 \times \mathbf{r}_3 = \alpha(\mathbf{e}_2 \times \mathbf{r}_2)$, where

$$\alpha = \frac{\sqrt{a_3^2 + b_4^2}}{a_2 + \sqrt{a_3^2 + b_4^2}}$$

in terms of the Denavit–Hartenberg notation, thereby making apparent that the last two columns of $\mathbf{J}_{21}$ are linearly dependent. Moreover, $\mathbf{e}_2 \times \mathbf{r}_2$ and, consequently, $\mathbf{e}_3 \times \mathbf{r}_3$ are parallel to $\mathbf{n}_{23}$, the last two axes being capable of producing velocities of C only in the direction of $\mathbf{n}_{23}$. Hence, velocities of C in Π_{23} that are normal to $\mathbf{e}_1 \times \mathbf{r}_1$, i.e., along line $\mathcal{L}$, cannot be produced in this configuration, and the manipulator loses, again, to a first-order approximation, one degree of freedom. The set of infeasible velocities, then, is parallel to the line $\mathcal{L}$ of Fig. 5.3, whose direction is the geometric representation of the null space of $\mathbf{J}_{21}^T$. The singularity displayed in the foregoing figure, termed here the *elbow singularity*, pertains also to a manipulator with the architecture of Fig. 4.3. Notice that motions along $\mathcal{L}$ in the posture displayed in Fig. 5.3 are possible, but only in one direction, from C to O_2.

With regard to the wrist singularities, these were already studied when solving the orientation problem for the inverse displacement analysis of decoupled manipulators. Here, we study the same in light of the sub-Jacobian $\mathbf{J}_{12}$ of Eq. (5.20b). This sub-Jacobian obviously vanishes when the wrist is so configured that its three revolute axes are coplanar, which thus leads to

$$\mathbf{e}_4 \times \mathbf{e}_5 \cdot \mathbf{e}_6 = 0$$

Note that when studying the orientation problem of decoupled manipulators, we found that orthogonal wrists are singular when the sixth and fourth axes are aligned, in full agreement with the foregoing condition. Indeed, if these two axes are aligned, then $\mathbf{e}_4 = -\mathbf{e}_6$, and the above equation holds.

5.4.1 Manipulator Workspace

The workspace of spherical wrists for orientation tasks was discussed in Sect. 4.4.2. Here we focus on the workspaces of three-axis positioning manipulators in light of their singularities.

In order to gain insight into the problem, we study first the workspace of manipulators with the architecture of Fig. 4.3. Figures 5.2 and 5.3 show such a manipulator with point C at the limit of its positioning capabilities in one direction, i.e., at the boundary of its workspace. Moreover, with regard to the posture of

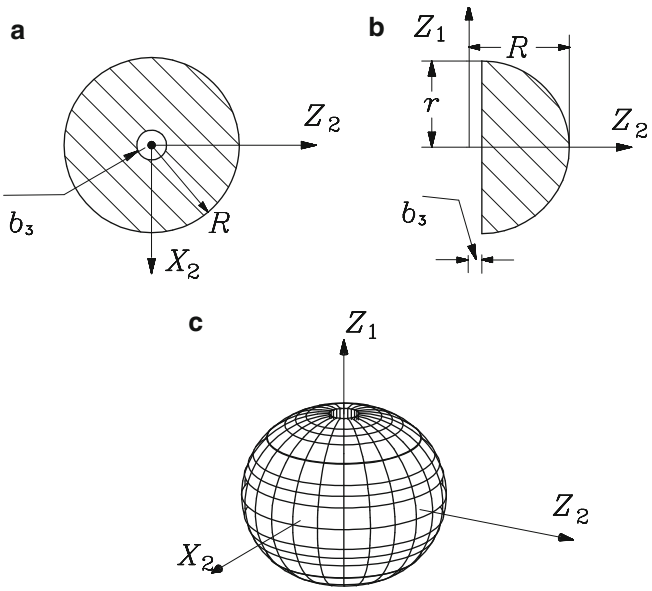


Fig. 5.4 Workspace of a Puma manipulator (**a**) top view; (**b**) cross-section; and (**c**) 3D view

Fig. 5.2, it is apparent that the first singularity is preserved if (a) point C moves on a line parallel to the first axis and intersecting the second axis; and (b) with the second and third joints locked, the first joint goes through a full turn. Under the second motion, the line of the first motion sweeps a circular cylinder whose axis is the first manipulator axis and with radius equal to b_3, the shoulder offset. This cylinder constitutes a part of the workspace boundary, the other part consisting of a spherical surface. Indeed, the second singularity is preserved if (a) with point C in the plane of the second and third axes, the second joint makes a full turn, thereby tracing a circle with center on $\mathcal{L}_2$, the plane of circle lying a distance b_3 from the first axis, with the circle radius $r = a_2 + \sqrt{a_3^2 + b_4^2}$; and (b) with point C still in the plane of the second and third joints, the first joint makes a full turn. Under the second motion, the circle generated by the first motion describes a sphere of radius $R = \sqrt{b_3^2 + r^2}$ because any point of that circle lies a distance R from the intersection of the first two axes. The point O_1, that coincides with O_2, thus becomes the center of the sphere, which is the second part of the workspace, as shown in Fig. 5.4.

The determination of the workspace boundaries of more general manipulators requires, obviously, more general approaches, like that proposed by Ceccarelli (1996). By means of an alternative approach, Ranjbaran et al. (1992) found the workspace boundary with the aid of the general characteristic equation of a three-revolute manipulator. This equation is a quartic polynomial, as displayed in Eq. (4.25). From the discussion of Sect. 4.4.1, it is apparent that at singularities, two

distinct roots of the IDP merge into a single one. This happens at points where the plot of the characteristic polynomial of Eq. (4.25) is tangent to the τ_3 axis, which occurs in turn at points where the derivative of this polynomial with respect to τ_3 vanishes. The condition for θ_3 to correspond to a point C on the boundary of the workspace is, then, that both the characteristic polynomial and its derivative with respect to τ_3 vanish concurrently. These two polynomials are displayed below:

$$P(\tau_3) \equiv R\tau_3^4 + S\tau_3^3 + T\tau_3^2 + U\tau_3 + V = 0 \tag{5.37a}$$

$$P'(\tau_3) \equiv 4R\tau_3^3 + 3S\tau_3^2 + 2T\tau_3 + U = 0 \tag{5.37b}$$

with coefficients R, S, T, U, and V defined in Eqs. (4.26a–e). From these equations and Eqs. (4.19d–f) and (4.20d–f), it is apparent that the foregoing coefficients are solely functions of the manipulator architecture and the Cartesian coordinates of point C. Moreover, from the same equations, it is apparent that the above coefficients are all *quadratic* in $\rho^2 \equiv x_C^2 + y_C^2$ and *quartic* in z_C. Thus, since the Cartesian coordinates x_C and y_C do not appear in the foregoing coefficients explicitly, the workspace is symmetric about the Z_1 axis, a result to be expected by virtue of the independence of singularities from angle θ_1. Hence, the workspace boundary is given by a function $f(\rho^2, z_C) = 0$ that can be derived by eliminating τ_3 from Eqs. (5.37a and b). This can be readily done by resorting to any elimination procedure, the simplest one being *dialytic elimination*, as discussed below.

In order to eliminate τ_3 from the above two equations, we proceed in two steps: In the first step, six additional polynomial equations are derived from Eqs. (5.37a and b) by multiplying the two sides of each of these equations by τ_3, τ_3^2, and τ_3^3, thereby obtaining a total of eight polynomial equations in τ_3, namely,

$$\begin{aligned}
R\tau_3^7 + S\tau_3^6 + T\tau_3^5 + U\tau_3^4 + V\tau_3^3 &= 0\\
4R\tau_3^6 + 3S\tau_3^5 + 2T\tau_3^4 + U\tau_3^3 &= 0\\
R\tau_3^6 + S\tau_3^5 + T\tau_3^4 + U\tau_3^3 + V\tau_3^2 &= 0\\
4R\tau_3^5 + 3S\tau_3^4 + 2T\tau_3^3 + U\tau_3^2 &= 0\\
R\tau_3^5 + S\tau_3^4 + T\tau_3^3 + U\tau_3^2 + V\tau_3 &= 0\\
4R\tau_3^4 + 3S\tau_3^3 + 2T\tau_3^2 + U\tau_3 &= 0\\
R\tau_3^4 + S\tau_3^3 + T\tau_3^2 + U\tau_3 + V &= 0\\
4R\tau_3^3 + 3S\tau_3^2 + 2T\tau_3 + U &= 0
\end{aligned}$$

In the second elimination step we write the above eight equations in *linear homogeneous form*, namely,

$$\mathbf{M}\boldsymbol{\tau}_3 = \mathbf{0} \tag{5.38a}$$

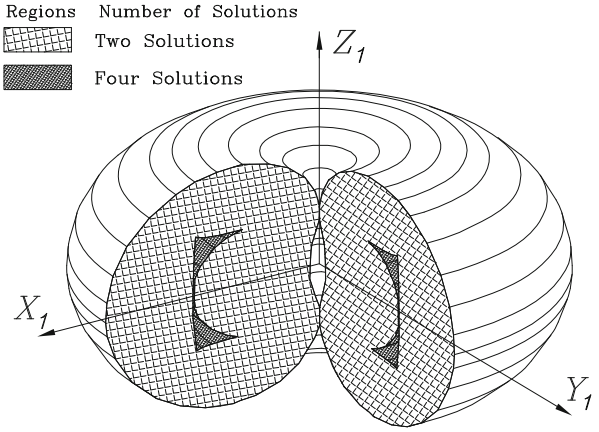


Fig. 5.5 The workspace of the manipulator of Figs. 4.15–4.17

with the 8×8 matrix $\mathbf{M}$ and the eight-dimensional vector $\boldsymbol{\tau}_3$ defined as

$$\mathbf{M} \equiv \begin{bmatrix} R & S & T & U & V & 0 & 0 & 0 \\ 0 & 4R & 3S & 2T & U & 0 & 0 & 0 \\ 0 & R & S & T & U & V & 0 & 0 \\ 0 & 0 & 4R & 3S & 2T & U & 0 & 0 \\ 0 & 0 & R & S & T & U & V & 0 \\ 0 & 0 & 0 & 4R & 3S & 2T & U & 0 \\ 0 & 0 & 0 & R & S & T & U & V \\ 0 & 0 & 0 & 0 & 4R & 3S & 2T & U \end{bmatrix}, \quad \boldsymbol{\tau}_3 = \begin{bmatrix} \tau_3^7 \\ \tau_3^6 \\ \tau_3^5 \\ \tau_3^4 \\ \tau_3^3 \\ \tau_3^2 \\ \tau_3 \\ 1 \end{bmatrix} \tag{5.38b}$$

It is now apparent that any feasible solution of Eq. (5.38a) must be nontrivial, and hence, $\mathbf{M}$ must be singular. The desired boundary equation is then derived from the singularity condition on $\mathbf{M}$, i.e.,

$$f(\rho^2, z_C) \equiv \det(\mathbf{M}) = 0 \tag{5.39}$$

As a matter of fact, function $f(\rho^2, z_C)$ of Eq. (5.39), known as the (polynomial) *resolvent* of Eqs. (5.37a and b), can be computed using computer algebra, upon invoking the procedure to obtain the discriminant of Eq. (5.37a).[4]

We used the foregoing procedure, with the help of computer algebra, to obtain a rendering of the workspace boundary of the manipulator of Figs. 4.15–4.17, the workspace thus obtained being displayed in Fig. 5.5. For the record, the resolvent

[4]Although a quartic polynomial has, properly speaking, four discriminants (Yang et al. 1996), which are defined as the central minors of matrix $\mathbf{M}$ of Eq. (5.39) when $P(\tau_3)$ is written in monic form—with leading coefficient equal to unity—the resolvent is sometimes referred to as *the* discriminant in question.

of this manipulator, given in Eq. (5.39), turned out to be a 16th-degree bivariate polynomial in ρ and z_C, involving only even powers. That is, the resolvent in question turns out to be a bivariate octic polynomial in ρ^2 and z_C^2. If we let $\sigma \equiv \rho^2$ and $\zeta \equiv z_C^2$, then

$$\begin{aligned} f(\sigma,\zeta) = {} & \zeta^8+(-2+\sigma)\zeta^7+(7\sigma^2-32\sigma+27)\zeta^6+(7\sigma^3-54\sigma^2+97\sigma-42)\zeta^5 \\ & +(35\sigma^4-400\sigma^3+1210\sigma^2-976\sigma+283)\zeta^4+(7\sigma^5-110\sigma^4+510\sigma^3 \\ & -684\sigma^2+123\sigma-70)\zeta^3+(7\sigma^6-144\sigma^5+965\sigma^4-2208\sigma^3+705\sigma^2 \\ & -272\sigma+83)\zeta^2+8(\sigma^2-4\sigma-1)(\sigma^5-22\sigma^4+154\sigma^3-328\sigma^2 \\ & -155\sigma+14)\zeta+(\sigma^4-24\sigma^3+190\sigma^2-552\sigma+17)(\sigma^2-4\sigma-1)^2=0 \end{aligned}$$

5.5 Acceleration Analysis of Serial Manipulators

The subject of this section is the computation of vector $\ddot{\boldsymbol{\theta}}$ of second joint-variable derivatives, also called the *joint accelerations*. This vector is computed from Cartesian position, velocity, and acceleration data. To this end, both sides of Eq. (5.9) are differentiated with respect to time, thus obtaining

$$\mathbf{J}\ddot{\boldsymbol{\theta}} = \dot{\mathbf{t}} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}} \tag{5.40}$$

and hence,

$$\ddot{\boldsymbol{\theta}} = \mathbf{J}^{-1}(\dot{\mathbf{t}} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}) \tag{5.41}$$

From Eq. (5.40), it is apparent that the joint-acceleration vector is computed in exactly the same way as the joint-rate vector. In fact, the LU decomposition of $\mathbf{J}$ is the same in this case and hence, need not be recomputed. All that is needed is the solution of a lower- and an upper-triangular system, namely,

$$\mathbf{L}\mathbf{z} = \dot{\mathbf{t}} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}, \quad \mathbf{U}\ddot{\boldsymbol{\theta}} = \mathbf{z}$$

The two foregoing systems are solved first for $\mathbf{z}$ and then for $\ddot{\boldsymbol{\theta}}$ by forward and backward substitution, respectively. The first of the foregoing systems is solved with M_n'' multiplications and A_n'' additions; the second with M_n''' multiplications and A_n''' additions. These figures appear in Eqs. (5.12b and c). Thus, the total numbers of multiplications M_t and additions A_t that the forward and backward solutions of the aforementioned systems require are

$$M_t = n^2, \quad A_t = n(n-1) \tag{5.42}$$

In Eq. (5.40), the right-hand side comprises two terms, the first being the specified time-rate of change of the twist of the EE, or twist-rate, for brevity, which is readily available. The second term is not available and must be computed. This term involves the product of the time-derivative of $\mathbf{J}$ times the previously computed joint-rate vector. Hence, in order to evaluate the right-hand side of that equation, all that is further required is $\dot{\mathbf{J}}$. From Eq. (5.10a), one has

$$\dot{\mathbf{J}} = \begin{bmatrix} \dot{\mathbf{A}} \\ \dot{\mathbf{B}} \end{bmatrix}$$

where, from Eqs. (5.7a and b),

$$\dot{\mathbf{A}} = \begin{bmatrix} \dot{\mathbf{e}}_1 & \dot{\mathbf{e}}_2 & \cdots & \dot{\mathbf{e}}_n \end{bmatrix} \tag{5.43a}$$

$$\dot{\mathbf{B}} = \begin{bmatrix} \dot{\mathbf{u}}_1 & \dot{\mathbf{u}}_2 & \cdots & \dot{\mathbf{u}}_n \end{bmatrix} \tag{5.43b}$$

and $\mathbf{u}_i$ denotes $\mathbf{e}_i \times \mathbf{r}_i$, for $i = 1, 2, \ldots, n$. Moreover,

$$\dot{\mathbf{e}}_1 = \boldsymbol{\omega}_0 \times \mathbf{e}_1 = \mathbf{0} \tag{5.44a}$$

$$\dot{\mathbf{e}}_i = \boldsymbol{\omega}_{i-1} \times \mathbf{e}_i \equiv \boldsymbol{\omega}_i \times \mathbf{e}_i, \quad i = 2, 3, \ldots, n \tag{5.44b}$$

and

$$\dot{\mathbf{u}}_i = \dot{\mathbf{e}}_i \times \mathbf{r}_i + \mathbf{e}_i \times \dot{\mathbf{r}}_i, \quad i = 1, 2, \ldots, n \tag{5.44c}$$

Next, an expression for $\dot{\mathbf{r}}_i$ is derived by time-differentiating both sides of Eq. (5.6), which produces

$$\dot{\mathbf{r}}_i = \dot{\mathbf{a}}_i + \dot{\mathbf{a}}_{i+1} + \cdots + \dot{\mathbf{a}}_n, \quad i = 1, 2, \ldots n$$

Recalling Eq. (5.4), the above equation reduces to

$$\dot{\mathbf{r}}_i = \boldsymbol{\omega}_i \times \mathbf{a}_i + \boldsymbol{\omega}_{i+1} \times \mathbf{a}_{i+1} + \cdots + \boldsymbol{\omega}_n \times \mathbf{a}_n \tag{5.45}$$

Substitution of Eqs. (5.44) and (5.45) into Eqs. (5.43a and b) leads to

$$\dot{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}_1 \times \mathbf{e}_2 & \cdots & \boldsymbol{\omega}_{n-1} \times \mathbf{e}_n \end{bmatrix}$$

$$\dot{\mathbf{B}} = \begin{bmatrix} \mathbf{e}_1 \times \dot{\mathbf{r}}_1 & \boldsymbol{\omega}_{12} \times \mathbf{r}_2 + \mathbf{e}_2 \times \dot{\mathbf{r}}_2 & \cdots & \boldsymbol{\omega}_{n-1,n} \times \mathbf{r}_n + \mathbf{e}_n \times \dot{\mathbf{r}}_n \end{bmatrix}$$

with $\dot{\mathbf{r}}_k$ and $\boldsymbol{\omega}_{k,k+1}$ defined as

$$\dot{\mathbf{r}}_k \equiv \sum_{k}^{n} \boldsymbol{\omega}_i \times \mathbf{a}_i, \quad k = 1, \ldots, n \tag{5.46a}$$

$$\boldsymbol{\omega}_{k,k+1} \equiv \boldsymbol{\omega}_k \times \mathbf{e}_{k+1}, \quad k = 1, \ldots, n-1 \tag{5.46b}$$

The foregoing expressions are invariant and hence, valid in any coordinate frame. However, these expressions have to be incorporated into matrix $\dot{\mathbf{J}}$; then, the latter is to be multiplied by vector $\dot{\boldsymbol{\theta}}$, as indicated in Eq. (5.40). Thus, eventually all columns of both $\dot{\mathbf{A}}$ and $\dot{\mathbf{B}}$ will have to be represented in the same coordinate frame. Hence, coordinate transformations will have to be introduced in the foregoing matrix columns in order to have all of these represented in the same coordinate frame, say, the first one. We then have the expansion below:

$$\dot{\mathbf{J}}\dot{\boldsymbol{\theta}} = \dot{\theta}_1 \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{u}}_1 \end{bmatrix} + \dot{\theta}_2 \begin{bmatrix} \dot{\mathbf{e}}_2 \\ \dot{\mathbf{u}}_2 \end{bmatrix} + \cdots + \dot{\theta}_n \begin{bmatrix} \dot{\mathbf{e}}_n \\ \dot{\mathbf{u}}_n \end{bmatrix} \tag{5.47}$$

The right-hand side of Eq. (5.47) is computed recursively as described below in five steps, the number of operations required being included at the end of each step.

1. Compute $\{ [\boldsymbol{\omega}_i]_i \}_1^n$:

$$[\boldsymbol{\omega}_1]_1 \leftarrow \dot{\theta}_1 [\mathbf{e}_1]_1$$

For i = 1 to n − 1 do

$$[\boldsymbol{\omega}_{i+1}]_{i+1} \leftarrow \dot{\theta}_{i+1} [\mathbf{e}_{i+1}]_{i+1} + \mathbf{Q}_i^T [\boldsymbol{\omega}_i]_i$$

enddo $\qquad 8(n-1)\,M \;\&\; 5(n-1)\,A$

2. Compute $\{ [\dot{\mathbf{e}}_i]_i \}_1^n$:

$$[\dot{\mathbf{e}}_1]_1 \leftarrow [\mathbf{0}]_1$$

For i = 2 to n do

$$[\dot{\mathbf{e}}_i]_i \leftarrow [\boldsymbol{\omega}_i \times \mathbf{e}_i]_i$$

enddo $\qquad 0M \;\&\; 0A$

3. Compute $\{ [\dot{\mathbf{r}}_i]_i \}_1^n$:

$$[\dot{\mathbf{r}}_n]_n \leftarrow [\boldsymbol{\omega}_n \times \mathbf{a}_n]_n$$

For i = n − 1 to 1 do

$$[\dot{\mathbf{r}}_i]_i \leftarrow [\boldsymbol{\omega}_i \times \mathbf{a}_i]_i + \mathbf{Q}_i [\dot{\mathbf{r}}_{i+1}]_{i+1}$$

enddo $\qquad (14n-8)M \;\&\; (10n-7)A$

4. Compute $\{ [\dot{\mathbf{u}}_i]_i \}_1^n$ using the expression appearing in Eq. (5.44c):

$$[\dot{\mathbf{u}}_1]_1 \leftarrow [\mathbf{e}_1 \times \dot{\mathbf{r}}_1]_1$$

For i = 2 to n do

$$[\dot{\mathbf{u}}_i]_i \leftarrow [\dot{\mathbf{e}}_i \times \mathbf{r}_i + \mathbf{e}_i \times \dot{\mathbf{r}}_i]_i$$

enddo $\qquad 4(n-1)\,M \;\&\; 3(n-1)\,A$

5. Compute $\dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$:
   Let $\mathbf{v} \equiv \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$, which is a six-dimensional vector. A coordinate transformation of its two three-dimensional vector components will be implemented using the 6×6 matrices $\mathbf{U}_i$, which are defined as

$$\mathbf{U}_i \equiv \begin{bmatrix} \mathbf{Q}_i & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_i \end{bmatrix}$$

where $\mathbf{O}$ stands for the 3×3 zero matrix. Thus, the foregoing 6×6 matrices are block-diagonal, their diagonal blocks being simply the matrices $\mathbf{Q}_i$ introduced in Sect. 4.2. One then has the algorithm below:

$$[\,\mathbf{v}\,]_n \leftarrow \dot{\theta}_n \begin{bmatrix} \dot{\mathbf{e}}_n \\ \dot{\mathbf{u}}_n \end{bmatrix}_n$$

`For i = n - 1 to 1 do`

$$[\,\mathbf{v}\,]_i \leftarrow \dot{\theta}_i \begin{bmatrix} \dot{\mathbf{e}}_i \\ \dot{\mathbf{u}}_i \end{bmatrix}_i + \mathbf{U}_i [\,\mathbf{v}\,]_{i+1}$$

`enddo`

$$\dot{\mathbf{J}}\dot{\boldsymbol{\theta}} \leftarrow [\,\mathbf{v}\,]_1 \qquad 21(n-1) + 4\,M \;\&\; 13(n-1)\,A$$

thereby completing the computation of $\dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$.

The figures given above for the floating-point operations involved were obtained based on a few facts, namely,

1. It is recalled that $[\,\mathbf{e}_i\,]_i = [\,0,\ 0,\ 1\,]^T$. Moreover, if we let $[\,\mathbf{w}\,]_i = [\,w_x,\ w_y,\ w_z\,]^T$ be an arbitrary three-dimensional vector, then

$$[\,\mathbf{e}_i \times \mathbf{w}\,]_i = \begin{bmatrix} -w_y \\ w_x \\ 0 \end{bmatrix}$$

this product thus requiring zero multiplications and zero additions.

2. $[\,\dot{\mathbf{e}}_i\,]_i$, computed as in Eq. (5.44b), takes on the form $[\,\omega_y,\ -\omega_x,\ 0\,]^T$, where ω_x and ω_y are the X_i and Y_i components of $\boldsymbol{\omega}_i$. Moreover, let $[\,\mathbf{r}_i\,]_i = [\,x,\ y,\ z\,]^T$. Then

$$[\,\dot{\mathbf{e}}_i \times \mathbf{r}_i\,]_i = \begin{bmatrix} -z\omega_x \\ -z\omega_y \\ x\omega_x + y\omega_y \end{bmatrix}$$

and this product is computed with four multiplications and one addition.

3. As found in Sect. 5.3, any coordinate transformation from $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$, or vice versa, of any three-dimensional vector is computed with eight multiplications and four additions.

Thus, the total numbers of multiplications and additions required to compute $\dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$ in frame $\mathcal{F}_1$, denoted by M_J and A_J, respectively, are as shown below:

$$M_J = 47n - 37, \quad A_J = 31n - 28$$

Since the right-hand side of Eq. (5.40) involves the algebraic sum of two six-dimensional vectors, then, the total numbers of multiplications and additions needed to compute the aforementioned right-hand side, denoted by M_r and A_r, are

$$M_r = 47n - 37, \quad A_r = 31n - 22$$

These figures yield 245 multiplications and 164 additions for a six-revolute manipulator of arbitrary architecture. Finally, if the latter figures are added to those of Eq. (5.42), one obtains the numbers of multiplications and additions required for an acceleration resolution of a six-revolute manipulator of arbitrary architecture as

$$M_a = 281, \quad A_a = 194$$

Furthermore, for six-axis, decoupled manipulators, the operation counts of steps 1 and 2 above do not change. However, step 3 is reduced by 42 multiplications and 30 additions, whereas step 4 by 12 multiplications and 9 additions. Moreover, step 5 is reduced by 63 multiplications and 39 additions. With regard to the solution of Eq. (5.40) for $\ddot{\boldsymbol{\theta}}$, an additional reduction of *floating-point operations*, or flops, is obtained, for now we need only 18 multiplications and 12 additions to solve two systems of three equations with three unknowns, thereby saving 18 multiplications and 18 additions. Thus, the corresponding figures for such a manipulator, M'_a and A'_a, respectively, are

$$M'_a = 146, \quad A'_a = 98$$

5.6 Static Analysis of Serial Manipulators

In this section, the static analysis of a serial n-axis manipulator is undertaken, six-axis decoupled manipulators being treated as special cases. Let τ_i be either the torque acting at the ith revolute or the force acting at the ith prismatic pair. Moreover, let $\boldsymbol{\tau}$ be the n-dimensional vector of joint forces and torques, whose ith component is τ_i, whereas $\mathbf{w} = [\mathbf{n}^T, \mathbf{f}^T]^T$ denotes the wrench exerted by the environment on the EE, with $\mathbf{n}$ denoting the resultant moment and $\mathbf{f}$ the resultant force applied at point P of the end-effector of the manipulator of Fig. 5.1. Then, the power exerted on the manipulator by all forces and moments acting on the EE is

$$\Pi_E = \mathbf{w}^T \mathbf{t} = \mathbf{n}^T \boldsymbol{\omega} + \mathbf{f}^T \dot{\mathbf{p}}$$

whereas the power Π_J exerted on the manipulator by all joint motors is

$$\Pi_J = \boldsymbol{\tau}^T \dot{\boldsymbol{\theta}} \tag{5.48}$$

Under static, conservative conditions, there is neither power dissipation nor change in the kinetic energy of the manipulator, and hence, the two foregoing powers are equal, which is just a restatement of the *First Law of Thermodynamics* or, equivalently, a form of the *Principle of Virtual Work*, i.e.,

$$\mathbf{w}^T \mathbf{t} = \boldsymbol{\tau}^T \dot{\boldsymbol{\theta}} \tag{5.49a}$$

Upon substitution of Eq. (5.9) into Eq. (5.49a), we obtain

$$\mathbf{w}^T \mathbf{J} \dot{\boldsymbol{\theta}} = \boldsymbol{\tau}^T \dot{\boldsymbol{\theta}} \tag{5.49b}$$

which is a relation valid for arbitrary $\dot{\boldsymbol{\theta}}$. Under these conditions, if $\mathbf{J}$ is not singular, Eq. (5.49b) leads to

$$\mathbf{J}^T \mathbf{w} = \boldsymbol{\tau} \tag{5.50}$$

This equation relates the wrench acting on the EE with the joint forces and torques exerted by the actuators. Therefore, this equation finds applications in the *sensing* of the wrench $\mathbf{w}$ acting on the EE by means of torque sensors located at the revolute axes. These sensors measure the motor-supplied torques via the current flowing through the motor armatures, the sensor readouts being the joint torques—or forces, in the case of prismatic joints—$\{ \tau_k \}_1^n$, grouped in vector $\boldsymbol{\tau}$.

For a six-axis manipulator, in the absence of singularities, the foregoing equation can be readily solved for $\mathbf{w}$ in the form

$$\mathbf{w} = \mathbf{J}^{-T} \boldsymbol{\tau}$$

where $\mathbf{J}^{-T}$ stands for the inverse of $\mathbf{J}^T$. Thus, using the figures recorded in Eq. (5.18b), $\mathbf{w}$ can be computed from Eq. (5.50) with 127 multiplications and 100 additions for a manipulator of arbitrary architecture. However, if the manipulator is of the decoupled type, the Jacobian takes on the form appearing in Eq. (5.19), and hence, the foregoing computation can be performed in two steps, namely,

$$\begin{aligned} \mathbf{J}_{12}^T \mathbf{n}_w &= \boldsymbol{\tau}_w \\ \mathbf{J}_{21}^T \mathbf{f} &= \boldsymbol{\tau}_a - \mathbf{J}_{11}^T \mathbf{n}_w \end{aligned}$$

where $\mathbf{n}_w$ is the resultant moment acting on the end-effector when $\mathbf{f}$ is applied at the center of the wrist, while $\boldsymbol{\tau}$ has been partitioned as

$$\boldsymbol{\tau} \equiv \begin{bmatrix} \boldsymbol{\tau}_a \\ \boldsymbol{\tau}_w \end{bmatrix}$$

with $\boldsymbol{\tau}_a$ and $\boldsymbol{\tau}_w$ defined as the wrist and the arm torques, respectively. These two vectors are given, in turn, as

$$\boldsymbol{\tau}_a = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}, \quad \boldsymbol{\tau}_w = \begin{bmatrix} \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix}$$

Hence, the foregoing calculations, as pertaining to a six-axis, decoupled manipulator, are performed with 55 multiplications and 37 additions, which follows from a result that was derived in Sect. 5.2 and is summarized in Eq. (5.23).

In solving for the wrench acting on the EE from the above relations, the wrist equilibrium equation is first solved for $\mathbf{n}_w$, thus obtaining

$$\mathbf{n}_w = \mathbf{J}_{12}^{-T} \boldsymbol{\tau}_w \tag{5.51}$$

where $\mathbf{J}_{12}^{-T}$ stands for the inverse of $\mathbf{J}_{12}^{T}$, and is available in Eq. (5.29). Therefore,

$$\begin{aligned} \mathbf{n}_w &= \frac{1}{\Delta_{12}} \left[(\mathbf{e}_5 \times \mathbf{e}_6) \ (\mathbf{e}_6 \times \mathbf{e}_4) \ (\mathbf{e}_4 \times \mathbf{e}_5) \right] \boldsymbol{\tau}_w \\ &= \frac{1}{\Delta_{12}} [\, \tau_4 (\mathbf{e}_5 \times \mathbf{e}_6) + \tau_5 (\mathbf{e}_6 \times \mathbf{e}_4) + \tau_6 (\mathbf{e}_4 \times \mathbf{e}_5) \,] \end{aligned} \tag{5.52}$$

Now, if we let

$$\overline{\boldsymbol{\tau}}_a \equiv \boldsymbol{\tau}_a - \mathbf{J}_{11}^{T} \mathbf{n}_w \tag{5.53}$$

we have, from Eq. (5.28),

$$\mathbf{f} = \left[\mathbf{u}_2 \times \mathbf{u}_3 \ \mathbf{u}_3 \times \mathbf{u}_1 \ \mathbf{u}_1 \times \mathbf{u}_2 \right] \frac{\overline{\boldsymbol{\tau}}_a}{\Delta_{21}}$$

where

$$\mathbf{u}_i \equiv \mathbf{e}_i \times \mathbf{r}_i$$

or

$$\mathbf{f} = \frac{1}{\Delta_{21}} [\, \overline{\tau}_1 (\mathbf{u}_2 \times \mathbf{u}_3) + \overline{\tau}_2 (\mathbf{u}_3 \times \mathbf{u}_1) + \overline{\tau}_3 (\mathbf{u}_1 \times \mathbf{u}_2) \,] \tag{5.54}$$

5.7 Planar Manipulators

Shown in Fig. 5.6 is a three-axis planar manipulator. Note that in this case, the DH parameters b_i and α_i vanish, for $i = 1, 2, 3$, the nonvanishing parameters a_i being indicated in the same figure. Below we proceed with the displacement, velocity, acceleration, and static analyses of this manipulator. Here, we recall a few relations of planar mechanics that will be found useful in the discussion below.

A 2×2 matrix $\mathbf{A}$ can be partitioned either columnwise or rowwise, as shown below:

$$\mathbf{A} \equiv \left[\mathbf{a} \ \mathbf{b} \right] \equiv \begin{bmatrix} \mathbf{c}^T \\ \mathbf{d}^T \end{bmatrix}$$

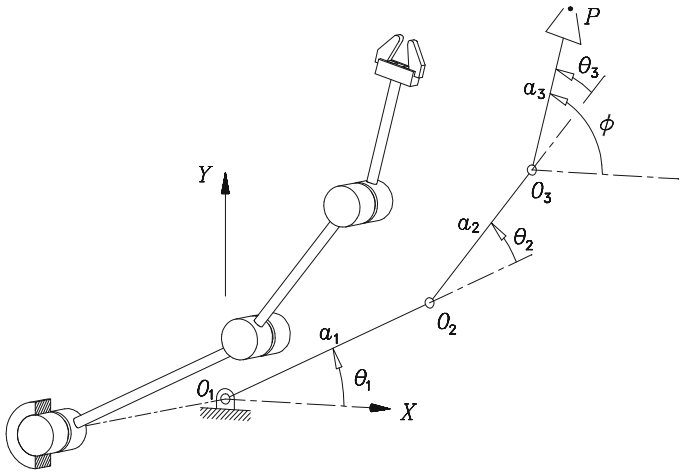


Fig. 5.6 Three-axis planar manipulator

where **a**, **b**, **c**, and **d** are all two-dimensional column vectors. Furthermore, let **E** be defined as an orthogonal matrix rotating two-dimensional vectors through an angle of 90° counterclockwise. Hence,

$$\mathbf{E} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{5.55}$$

We thus have

Fact 5.7.1.

$$\mathbf{E}^{-1} = \mathbf{E}^T = -\mathbf{E}$$

and hence,

Fact 5.7.2.

$$\mathbf{E}^2 = -\mathbf{1}$$

where **1** is the 2×2 identity matrix. Moreover,

Fact 5.7.3.

$$\det(\mathbf{A}) = -\mathbf{a}^T\mathbf{E}\mathbf{b} = \mathbf{b}^T\mathbf{E}\mathbf{a} = -\mathbf{c}^T\mathbf{E}\mathbf{d} = \mathbf{d}^T\mathbf{E}\mathbf{c}$$

and

Fact 5.7.4.

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})}\begin{bmatrix}\mathbf{b}^T \\ -\mathbf{a}^T\end{bmatrix}\mathbf{E} = \frac{1}{\det(\mathbf{A})}\mathbf{E}\begin{bmatrix}-\mathbf{d} & \mathbf{c}\end{bmatrix}$$

5.7.1 Displacement Analysis

The inverse displacement analysis of the manipulator at hand now consists in determining the values of angles θ_i, for $i = 1, 2, 3$, that will place the end-effector so that its operation point P will be positioned at the prescribed Cartesian coordinates x, y and be oriented at a given angle ϕ with the X axis of Fig. 5.6. Note that this manipulator can be considered as decoupled, for the end-effector can be placed at the desired pose by first positioning point O_3 with the aid of the first two joints and then orienting it with the third joint only. We then solve for the joint angles in two steps, one for positioning and one for orienting.

We now have, from the geometry of Fig. 5.6,

$$a_1 c_1 + a_2 c_{12} = x$$
$$a_1 s_1 + a_2 s_{12} = y$$

where x and y denote the Cartesian coordinates of point O_3, while c_{12} and s_{12} stand for $\cos(\theta_1 + \theta_2)$ and $\sin(\theta_1 + \theta_2)$, respectively. We have thus derived two equations for the two unknown angles, from which we can determine these angles in various ways. For example, we can solve the problem using a semigraphical approach similar to that of Sect. 9.4.

Indeed, from the two foregoing equations we can eliminate both c_{12} and s_{12} by solving for the second terms of the left-hand sides of those equations, namely,

$$a_2 c_{12} = x - a_1 c_1 \tag{5.56a}$$
$$a_2 s_{12} = y - a_1 s_1 \tag{5.56b}$$

If both sides of the above two equations are now squared, then added, and the ensuing sum is equated to a_2^2, we obtain, after simplification, a linear equation in c_1 and s_1 that represents a line $\mathcal{L}$ in the c_1-s_1 plane:

$$\mathcal{L}: \qquad -a_1^2 + a_2^2 + 2a_1 x c_1 + 2a_1 y s_1 - (x^2 + y^2) = 0 \tag{5.57}$$

Apparently, the two foregoing variables are constrained by a quadratic equation defining a circle $\mathcal{C}$ in the same plane:

$$\mathcal{C}: \qquad c_1^2 + s_1^2 = 1$$

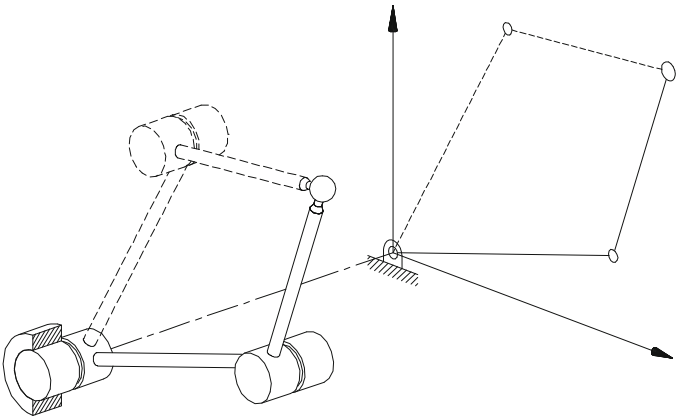


Fig. 5.7 The two real solutions of a planar manipulator

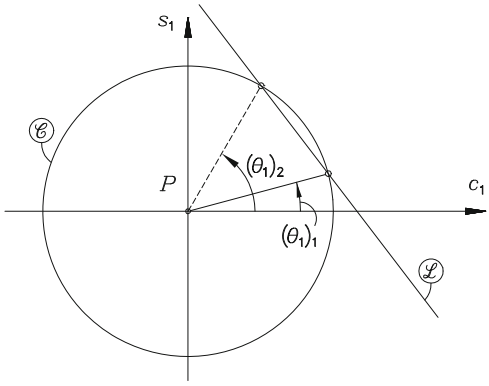


Fig. 5.8 The two real values of θ_1 giving the two postures depicted in Fig. 5.7

which has a unit radius and is centered at the origin of the c_1-s_1 plane. The real roots of interest are then obtained as the intersections of $\mathcal{L}$ and $\mathcal{C}$. Thus, the problem can admit (a) two real and distinct roots, if the line and the circle intersect; (b) one repeated root if the line is tangent to the circle; and (c) no real root if the line does not intersect the circle (Fig. 5.7).

With c_1 and s_1 known, angle θ_1 is fully determined. Note that the two real intersections of $\mathcal{L}$ with $\mathcal{C}$ provide each one value of θ_1, as depicted in Fig. 5.8.

Once θ_1 and θ_2 are available, θ_3 is readily derived from the geometry of Fig. 5.6, namely,

$$\theta_3 = \phi - (\theta_1 + \theta_2)$$

and hence, each pair of $(\theta_1,\ \theta_2)$ values yields one single value for θ_3. Since we have two such pairs, the problem admits two real solutions.

5.7.2 *Velocity Analysis*

Velocity analysis is most easily accomplished if the general velocity relations derived in Sect. 5.2 are recalled and adapted to planar manipulators. Thus we have, as in Eq. (5.9),

$$\mathbf{J}\dot{\boldsymbol{\theta}} = \mathbf{t} \tag{5.58a}$$

where now,

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \mathbf{e}_3 \times \mathbf{r}_3 \end{bmatrix}, \quad \dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{p}} \end{bmatrix} \tag{5.58b}$$

and $\{\,\mathbf{r}_i\,\}_1^3$ are defined as in Eq. (5.6), i.e., as the vectors directed from O_i to P. As in the previous subsection, we assume here that the manipulator moves in the X–Y plane, and hence, all revolute axes are parallel to the Z axis, vectors $\mathbf{e}_i$ and $\mathbf{r}_i$, for $i = 1, 2, 3$, thus taking on the forms

$$\mathbf{e}_1 = \mathbf{e}_2 = \mathbf{e}_3 = \mathbf{e} \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix}$$

with $\mathbf{t}$ reducing to

$$\mathbf{t} = \begin{bmatrix} 0 & 0 & \dot{\phi} & \dot{x}_P & \dot{y}_P & 0 \end{bmatrix}^T \tag{5.58c}$$

in which $\dot{x}_P$ and $\dot{y}_P$ denote the components of the velocity of P. Thus,

$$\mathbf{e}_i \times \mathbf{r}_i = \begin{bmatrix} -y_i \\ x_i \\ 0 \end{bmatrix}$$

and hence, the foregoing cross product can be expressed as

$$\mathbf{e}_i \times \mathbf{r}_i = \begin{bmatrix} \mathbf{E}\mathbf{s}_i \\ 0 \end{bmatrix}$$

where $\mathbf{E}$ was defined in Eq. (5.55) and $\mathbf{s}_i$ is the two-dimensional projection of $\mathbf{r}_i$ onto the X–Y plane of motion, i.e., $\mathbf{s}_i \equiv \begin{bmatrix} x_i & y_i \end{bmatrix}^T$. Equation (5.58a) thus reduces to

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 1 \\ \mathbf{E}\mathbf{s}_1 & \mathbf{E}\mathbf{s}_2 & \mathbf{E}\mathbf{s}_3 \\ 0 & 0 & 0 \end{bmatrix} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \mathbf{0} \\ \dot{\phi} \\ \dot{\mathbf{p}} \\ 0 \end{bmatrix} \tag{5.59}$$

where $\mathbf{0}$ is the two-dimensional zero vector and $\dot{\mathbf{p}}$ is now reduced to $\dot{\mathbf{p}} \equiv [\,\dot{x},\ \dot{y}\,]^T$. In summary, then, by working only with the three nontrivial equations of Eq. (5.59), we can represent the velocity relation using a 3×3 Jacobian in Eq. (5.58a). To this end, we redefine $\mathbf{J}$ and $\mathbf{t}$ as

$$\mathbf{J} \equiv \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{Es}_1 & \mathbf{Es}_2 & \mathbf{Es}_3 \end{bmatrix}, \quad \mathbf{t} \equiv \begin{bmatrix} \dot{\phi} \\ \dot{\mathbf{p}} \end{bmatrix} \tag{5.60}$$

The velocity resolution of this manipulator thus reduces to solving for the three joint rates from Eq. (5.58a), with $\mathbf{J}$ and $\mathbf{t}$ defined as in Eq. (5.60), which thus leads to the system below:

$$\begin{bmatrix} 1 & 1 & 1 \\ \mathbf{Es}_1 & \mathbf{Es}_2 & \mathbf{Es}_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\mathbf{p}} \end{bmatrix} \tag{5.61}$$

Solving for $\{\,\dot{\theta}_i\,\}_1^3$ is readily done by first reducing the system of equations appearing in Eq. (5.58a) to one of two equations in two unknowns by resorting to Gaussian elimination. Indeed, if the first scalar equation of Eq. (5.61) is multiplied by $\mathbf{Es}_1$ and the product is subtracted from the two-dimensional vector equation, we obtain

$$\begin{bmatrix} 1 & 1 & 1 \\ \mathbf{0} & \mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1) & \mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\mathbf{p}} - \dot{\phi}\mathbf{Es}_1 \end{bmatrix} \tag{5.62}$$

from which a reduced system of two equations in two unknowns is readily obtained, namely,

$$\begin{bmatrix} \mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1) & \mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1) \end{bmatrix} \begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \dot{\mathbf{p}} - \dot{\phi}\mathbf{Es}_1 \tag{5.63}$$

The system of equations (5.63) can be readily solved if Fact 5.7.4 is recalled, namely,

$$\begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} -(\mathbf{s}_3 - \mathbf{s}_1)^T\mathbf{E} \\ (\mathbf{s}_2 - \mathbf{s}_1)^T\mathbf{E} \end{bmatrix} \mathbf{E}(\dot{\mathbf{p}} - \dot{\phi}\mathbf{Es}_1)$$

$$= \frac{1}{\Delta} \begin{bmatrix} (\mathbf{s}_3 - \mathbf{s}_1)^T(\dot{\mathbf{p}} - \dot{\phi}\mathbf{Es}_1) \\ -(\mathbf{s}_2 - \mathbf{s}_1)^T(\dot{\mathbf{p}} - \dot{\phi}\mathbf{Es}_1) \end{bmatrix}$$

where Δ is the determinant of the 2×2 matrix coefficient of Eq. (5.63), i.e.,

$$\Delta \equiv \det(\begin{bmatrix} \mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1) & \mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1) \end{bmatrix}) \equiv -(\mathbf{s}_2 - \mathbf{s}_1)^T\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1) \tag{5.64}$$

We thus have

$$\dot{\theta}_2 = -\frac{(\mathbf{s}_3 - \mathbf{s}_1)^T(\dot{\mathbf{p}} - \dot{\phi}\mathbf{E}\mathbf{s}_1)}{(\mathbf{s}_2 - \mathbf{s}_1)^T\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1)} \tag{5.65a}$$

$$\dot{\theta}_3 = \frac{(\mathbf{s}_2 - \mathbf{s}_1)^T(\dot{\mathbf{p}} - \dot{\phi}\mathbf{E}\mathbf{s}_1)}{(\mathbf{s}_2 - \mathbf{s}_1)^T\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1)} \tag{5.65b}$$

Further, $\dot{\theta}_1$ is computed from the first scalar equation of Eq. (5.61), i.e.,

$$\dot{\theta}_1 = \dot{\phi} - (\dot{\theta}_2 + \dot{\theta}_3) \tag{5.65c}$$

thereby completing the velocity analysis.

The foregoing calculations are summarized below in algorithmic form, with the corresponding numbers of multiplications and additions indicated at each stage. In those numbers, we have taken into account that a multiplication of $\mathbf{E}$ by any two-dimensional vector incurs no computational cost, but rather a simple rearrangement of the entries of this vector, with a reversal of one sign.

1. $\mathbf{d}_{21} \leftarrow \mathbf{s}_2 - \mathbf{s}_1$ $\quad 0M + 2A$
2. $\mathbf{d}_{31} \leftarrow \mathbf{s}_3 - \mathbf{s}_1$ $\quad 0M + 2A$
3. $\Delta \leftarrow \mathbf{d}_{31}^T\mathbf{E}\mathbf{d}_{21}$ $\quad 2M + 1A$
4. $\mathbf{u} \leftarrow \dot{\mathbf{p}} - \dot{\phi}\mathbf{E}\mathbf{s}_1$ $\quad 2M + 2A$
5. $\mathbf{u} \leftarrow \mathbf{u}/\Delta$ $\quad 2M + 0A$
6. $\dot{\theta}_2 \leftarrow \mathbf{u}^T\mathbf{d}_{31}$ $\quad 2M + 1A$
7. $\dot{\theta}_3 \leftarrow -\mathbf{u}^T\mathbf{d}_{21}$ $\quad 2M + 1A$
8. $\dot{\theta}_1 \leftarrow \dot{\phi} - \dot{\theta}_2 - \dot{\theta}_3$ $\quad 0M + 2A$

The complete calculation of joint rates thus consumes only $10M$ and $11A$, which represents a savings of about 67% of the computations involved if Gaussian elimination is applied without regarding the algebraic structure of the Jacobian $\mathbf{J}$ and its kinematic and geometric significance. In fact, the solution of an arbitrary system of three equations in three unknowns requires, from Eq. (5.18a), 28 additions and 23 multiplications. If the cost of calculating the right-hand side is added, namely, $4A$ and $6M$, a total of $32A$ and $29M$ is required to solve for the joint rates if straightforward Gaussian elimination is used.

5.7.3 *Acceleration Analysis*

The calculation of the joint accelerations needed to produce a given twist rate of the EE is readily accomplished by differentiating both sides of Eq. (5.58a), with definitions (5.60), i.e.,

$$\mathbf{J}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\theta}} = \dot{\mathbf{t}}$$

from which we readily derive a system of equations similar to Eq. (5.58a) with $\ddot{\boldsymbol{\theta}}$ as unknown, namely,

$$\mathbf{J}\ddot{\boldsymbol{\theta}} = \dot{\mathbf{t}} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$$

where

$$\dot{\mathbf{J}} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{E}\dot{\mathbf{s}}_1 & \mathbf{E}\dot{\mathbf{s}}_2 & \mathbf{E}\dot{\mathbf{s}}_3 \end{bmatrix}, \quad \ddot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}, \quad \dot{\mathbf{t}} \equiv \begin{bmatrix} \ddot{\phi} \\ \ddot{\mathbf{p}} \end{bmatrix}$$

and

$$\begin{aligned} \dot{\mathbf{s}}_3 &= (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\mathbf{E}\mathbf{a}_3 \\ \dot{\mathbf{s}}_2 &= \dot{\mathbf{a}}_2 + \dot{\mathbf{s}}_3 = (\dot{\theta}_1 + \dot{\theta}_2)\mathbf{E}\mathbf{a}_2 + \dot{\mathbf{s}}_3 \\ \dot{\mathbf{s}}_1 &= \dot{\mathbf{a}}_1 + \dot{\mathbf{s}}_2 = \dot{\theta}_1\mathbf{E}\mathbf{a}_1 + \dot{\mathbf{s}}_2 \end{aligned}$$

Now we can proceed by Gaussian elimination to solve for the joint accelerations in exactly the same manner as in Sect. 5.7.2, thereby obtaining the counterpart of Eq. (5.63), namely,

$$\begin{bmatrix} \mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1) & \mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1) \end{bmatrix} \begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \mathbf{w} \tag{5.66a}$$

with $\mathbf{w}$ defined as

$$\mathbf{w} \equiv \ddot{\mathbf{p}} - \mathbf{E}(\dot{\theta}_1\dot{\mathbf{s}}_1 + \dot{\theta}_2\dot{\mathbf{s}}_2 + \dot{\theta}_3\dot{\mathbf{s}}_3 + \ddot{\phi}\mathbf{s}_1) \tag{5.66b}$$

and hence, similar to Eqs. (5.65a–c), one has

$$\ddot{\theta}_2 = \frac{(\mathbf{s}_3 - \mathbf{s}_1)^T\mathbf{w}}{\Delta} \tag{5.67a}$$

$$\ddot{\theta}_3 = -\frac{(\mathbf{s}_2 - \mathbf{s}_1)^T\mathbf{w}}{\Delta} \tag{5.67b}$$

$$\ddot{\theta}_1 = \ddot{\phi} - (\dot{\theta}_2 + \dot{\theta}_3) \tag{5.67c}$$

Below we summarize the foregoing calculations in algorithmic form, indicating the numbers of operations required at each stage.

1. $\dot{\mathbf{s}}_3 \leftarrow (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\mathbf{E}\mathbf{a}_3$	$2M$ & $2A$
2. $\dot{\mathbf{s}}_2 \leftarrow (\dot{\theta}_1 + \dot{\theta}_2)\mathbf{E}\mathbf{a}_2 + \dot{\mathbf{s}}_3$	$2M$ & $3A$
3. $\dot{\mathbf{s}}_1 \leftarrow \dot{\theta}_1\mathbf{E}\mathbf{a}_1 + \dot{\mathbf{s}}_2$	$2M$ & $2A$
4. $\mathbf{w} \leftarrow \ddot{\mathbf{p}} - \mathbf{E}(\dot{\theta}_1\dot{\mathbf{s}}_1 + \dot{\theta}_2\dot{\mathbf{s}}_2 + \dot{\theta}_3\dot{\mathbf{s}}_3 + \ddot{\phi}\mathbf{s}_1)$	$8M$ & $8A$
5. $\mathbf{w} \leftarrow \mathbf{w}/\Delta$	$2M + 0A$
6. $\ddot{\theta}_2 \leftarrow \mathbf{w}^T\mathbf{d}_{31}$	$2M + 1A$
7. $\ddot{\theta}_3 \leftarrow -\mathbf{w}^T\mathbf{d}_{21}$	$2M + 1A$
8. $\ddot{\theta}_1 \leftarrow \ddot{\phi} - (\ddot{\theta}_2 + \ddot{\theta}_3)$	$0M + 2A$

where $\mathbf{d}_{21}$, $\mathbf{d}_{31}$, and Δ are available from velocity calculations. The joint accelerations thus require a total of 20 multiplications and 19 additions. These figures represent substantial savings when compared with the numbers of operations required if plain Gaussian elimination were used, namely, 33 multiplications and 35 additions.

It is noteworthy that in the foregoing algorithm, we have replaced neither the sum $\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3$ nor $\dot{\theta}_1 \mathbf{E}(\mathbf{s}_1 + \mathbf{s}_2 + \mathbf{s}_3)$ by ω and correspondingly, by $\dot{\mathbf{p}}$, because in path tracking, there is no perfect match between joint and Cartesian variables. In fact, joint-rate and joint-acceleration calculations are needed in feedback control schemes to estimate the position, velocity, and acceleration errors by proper corrective actions.

5.7.4 Static Analysis

Here we assume that the environment exerts a planar wrench on the EE of the manipulator appearing in Fig. 5.6. In accordance with the definition of the planar twist in Sect. 5.7.2, Eq. (5.60), the planar wrench is now defined as

$$\mathbf{w} \equiv \begin{bmatrix} n \\ \mathbf{f} \end{bmatrix} \tag{5.68}$$

where n is the scalar couple acting on the EE and $\mathbf{f}$ is the two-dimensional force acting at the operation point P of the EE. If additionally, we denote by $\boldsymbol{\tau}$ the three-dimensional vector of joint torques, the planar counterpart of Eq. (5.50) follows, i.e.,

$$\mathbf{J}^T \mathbf{w} = \boldsymbol{\tau} \tag{5.69}$$

where

$$\mathbf{J}^T = \begin{bmatrix} 1 & (\mathbf{E}\mathbf{s}_1)^T \\ 1 & (\mathbf{E}\mathbf{s}_2)^T \\ 1 & (\mathbf{E}\mathbf{s}_3)^T \end{bmatrix}$$

Now, in order to solve for the wrench $\mathbf{w}$ acting on the end-effector, given the joint torques $\boldsymbol{\tau}$ and the posture of the manipulator, we can still apply our compact Gaussian-elimination scheme, as introduced in Sect. 5.7.2. To this end, we subtract the first scalar equation from the second and the third scalar equations of Eq. (5.69), which renders the foregoing system in the form

$$\begin{bmatrix} 1 & (\mathbf{E}\mathbf{s}_1)^T \\ 0 & [\mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1)]^T \\ 0 & [\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1)]^T \end{bmatrix} \begin{bmatrix} n \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 - \tau_1 \\ \tau_3 - \tau_1 \end{bmatrix}$$

Thus, the last two equations have been decoupled from the first one, which allows us to solve them separately, i.e., we have reduced the system to one of two equations in two unknowns, namely,

$$\begin{bmatrix} [\mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1)]^T \\ [\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1)]^T \end{bmatrix} \mathbf{f} = \begin{bmatrix} \tau_2 - \tau_1 \\ \tau_3 - \tau_1 \end{bmatrix} \tag{5.70}$$

from which we readily obtain

$$\mathbf{f} = \begin{bmatrix} [\mathbf{E}(\mathbf{s}_2 - \mathbf{s}_1)]^T \\ [\mathbf{E}(\mathbf{s}_3 - \mathbf{s}_1)]^T \end{bmatrix}^{-1} \begin{bmatrix} \tau_2 - \tau_1 \\ \tau_3 - \tau_1 \end{bmatrix} \tag{5.71}$$

and hence, upon expansion of the above inverse,

$$\mathbf{f} = \frac{1}{\Delta} \left[(\tau_2 - \tau_1)(\mathbf{s}_3 - \mathbf{s}_1) - (\tau_3 - \tau_1)(\mathbf{s}_2 - \mathbf{s}_1) \right] \tag{5.72}$$

where Δ is exactly as defined in Eq. (5.64). Finally, the resultant moment n acting on the end-effector is readily calculated from the first scalar equation of Eq. (5.69), namely, as

$$n = \tau_1 + \mathbf{s}_1^T \mathbf{E}\mathbf{f}$$

thereby completing the static analysis of the manipulator under study. A quick analysis of computational costs shows that the foregoing solution needs $8M$ and $6A$, or a savings of about 70% if straightforward Gaussian elimination is applied.

5.8 Kinetostatic Performance Indices

Chapters 6 and 7 do not depend on this section, which can thus be skipped in an introductory course based on the first half of the book. We have included this section because (a) it is a simple matter to render the section self-contained, while introducing the concept of *condition number* and its relevance in robotics; (b) kinetostatic performance can be studied with the background of the material included up to this section; and (c) kinetostatic performance is becoming increasingly relevant as a design criterion and as a figure of merit in robot control.

A *kinetostatic performance index* of a robotic mechanical system is a scalar quantity that measures how well the system behaves with regard to force and motion transmission, the latter being understood in the differential sense, i.e., at the velocity level. Now, a kinetostatic performance index, or kinetostatic index for brevity, may be needed to assess the performance of a robot at the design stage, in which case we need a posture-independent index. In this case, the index becomes a function of the robot architecture only. If, on the other hand, we want to assess the performance

of a *given* robot while performing a task, what we need is a posture-dependent index. This difference is often overlooked in the robotics literature, although it is extremely important. Moreover, while performance indices can be defined for all kinds of robotic mechanical systems, we focus here on those associated with serial manipulators, which are the ones studied most intensively.

Among the various performance indices that have been proposed, one can cite the concept of *service angle*, first introduced by Vinogradov et al. (1971), and the *conditioning* of robotic manipulators, as proposed by Yang and Lai (1985). Yoshikawa (1985), in turn, introduced the concept of *manipulability*, which is defined as the square root of the determinant of the product of the manipulator Jacobian by its transpose. Paul and Stevenson (1983) used the absolute value of the determinant of the Jacobian to assess the kinematic performance of spherical wrists. Note that Yoshikawa's manipulability is identical to the absolute value of the determinant of the Jacobian, and hence, the latter coincides with Paul and Stevenson's performance index. It should be pointed out that these indices were defined for control purposes and hence, are posture-dependent. Germane to these concepts is that of *dextrous workspace*, introduced by Kumar and Waldron (1981), and used for geometric optimization by Vijaykumar et al. (1986). Although the concepts of service angle and manipulability are apparently different, they touch upon a common underlying issue, namely, the kinematic, or alternatively, the static performance of a manipulator from an accuracy viewpoint. For this reason, we refer to these indices generically as *kinetostatic*.

What is at stake when discussing the manipulability of a robotic manipulator is a measure of the *invertibility* of the associated Jacobian matrix, since this is required for velocity and force-feedback control. One further performance index is based on the *condition number* of the Jacobian, which was first used by Salisbury and Craig (1982) to design mechanical fingers. Here, we shall use this concept to define the *kinetostatic conditioning index* of the manipulator. For the sake of brevity, we devote the discussion below to only two indices, namely, manipulability and conditioning. Prior to discussing these indices, we recall a few facts from linear algebra.

Although the concepts discussed here are equally applicable to square and rectangular matrices, we shall focus on the former. First, we give a geometric interpretation of the mapping induced by a $n \times n$ matrix $\mathbf{A}$. Here, we do not assume any particular structure of $\mathbf{A}$, which can thus be totally arbitrary. However, by invoking the *polar-decomposition theorem* (Strang 1988), we can factor $\mathbf{A}$ as

$$\mathbf{A} \equiv \mathbf{RU} \equiv \mathbf{VR} \tag{5.73}$$

where $\mathbf{R}$ is orthogonal, although not necessarily proper, while $\mathbf{U}$ and $\mathbf{V}$ are both at least positive-semidefinite. Moreover, if $\mathbf{A}$ is nonsingular, then $\mathbf{U}$ and $\mathbf{V}$ are both positive-definite, and $\mathbf{R}$ is unique. Apparently,

$$\mathbf{A}^T\mathbf{A} = \mathbf{U}^2 \quad \text{or} \quad \mathbf{A}\mathbf{A}^T = \mathbf{V}^2 \tag{5.74}$$

and hence, $\mathbf{U}\,(\mathbf{V})$ can be readily determined as the positive-semidefinite or correspondingly, positive-definite *square root* of the product $\mathbf{A}^T\mathbf{A}\,(\mathbf{A}\mathbf{A}^T)$, which is necessarily positive-semidefinite at least; it is, in fact, positive-definite if $\mathbf{A}$ is nonsingular. We recall here that the square root of arbitrary matrices was briefly discussed in Sect. 2.3.6. The square root of a positive-semidefinite matrix can be most easily understood if that matrix is assumed to be in diagonal form, which is possible because such a matrix is necessarily symmetric, and every symmetric matrix is diagonalizable. The matrix at hand being positive-semidefinite, its eigenvalues are nonnegative, and hence, their square roots are all real. The positive-semidefinite square root of interest is, then, readily obtained as the diagonal matrix whose nontrivial entries are the nonnegative square roots of the above-mentioned eigenvalues. With $\mathbf{U}$ or $\mathbf{V}$ determined, $\mathbf{R}$ can be found uniquely only if $\mathbf{A}$ is nonsingular, in which case $\mathbf{U}$ and $\mathbf{V}$ are positive-definite. If this is the case, then we have

$$\mathbf{R} = \mathbf{A}\mathbf{U}^{-1} = \mathbf{V}^{-1}\mathbf{A} \tag{5.75a}$$

It is a simple matter to show that $\mathbf{U}$ and $\mathbf{V}$ are related by a similarity transformation, namely,

$$\mathbf{V} = \mathbf{R}\mathbf{U}\mathbf{R}^T \tag{5.75b}$$

Now, as a consequence of the above relation between $\mathbf{U}$ and $\mathbf{V}$, both matrices share the same set of nonnegative eigenvalues $\{\sigma_i\}_1^n$, which are termed the *singular values* of the given matrix $\mathbf{A}$. Furthermore, if the eigenvectors of $\mathbf{U}$ are denoted by $\{\mathbf{u}_i\}_1^n$ and those of $\mathbf{V}$ by $\{\mathbf{v}_i\}_1^n$, then the two sets are related by a similarity transformation as well:

$$\mathbf{v}_i = \mathbf{R}\mathbf{u}_i\,, \quad i = 1, \ldots, n \tag{5.76}$$

Now, let vector $\mathbf{x}$ be mapped by $\mathbf{A}$ into $\mathbf{z}$, i.e.,

$$\mathbf{z} = \mathbf{A}\mathbf{x} \equiv \mathbf{R}\mathbf{U}\mathbf{x} \tag{5.77a}$$

Moreover, let

$$\mathbf{y} \equiv \mathbf{U}\mathbf{x} \tag{5.77b}$$

and hence, we have a concatenation of mappings: $\mathbf{U}$ maps $\mathbf{x}$ into $\mathbf{y}$, while $\mathbf{R}$ maps $\mathbf{y}$ into $\mathbf{z}$. Thus, by virtue of the nature of matrices $\mathbf{R}$ and $\mathbf{U}$, the latter maps the unit n-dimensional ball into a n-axis ellipsoid whose semiaxis lengths bear the ratios of the eigenvalues of $\mathbf{U}$. Moreover, $\mathbf{R}$ maps this ellipsoid into another one with identical semiaxes, except that it is rotated about its center or reflected, depending upon whether $\mathbf{R}$ is proper or improper orthogonal. The eigenvalues of $\mathbf{U}$ or, for that matter, those of $\mathbf{V}$, are thus nothing but the *singular values* of $\mathbf{A}$. Yoshikawa (1985)

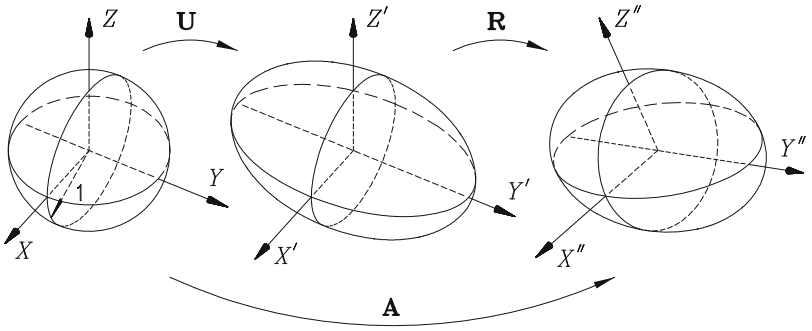


Fig. 5.9 Geometric representation of mapping induced by matrix **A**

explained the foregoing relations resorting to the *singular-value decomposition theorem*. We prefer to invoke the polar-decomposition theorem instead, because of the geometric nature of the latter, as opposed to the former, which is of an algebraic nature—it is based on a diagonalization of either **U** or **V**, which is really not needed.

We illustrate the two mappings **U** and **R** in Fig. 5.9, where we orient the X, Y, and Z axes along the three eigenvectors of **U**. Therefore, the semiaxes of the ellipsoid are oriented as the eigenvectors of **U** as well. If **A** is singular, then the ellipsoid degenerates into one with at least one vanishing semiaxis. On the other hand, if matrix **A** is *isotropic*, i.e., if all its singular values are identical, then it maps the unit ball into another ball, either enlarged or shrunken.

For our purposes, we can regard the Jacobian of a serial manipulator as mapping the unit ball in the space of joint rates into a rotated or reflected ellipsoid in the space of Cartesian velocities, or twists. Now, let us assume that the polar decomposition of **J** is given by **R** and **U**, the manipulability μ of the robot under study thus becoming

$$\mu \equiv |\det(\mathbf{J})| \equiv |\det(\mathbf{R})||\det(\mathbf{U})| \tag{5.78a}$$

Since **R** is orthogonal, the absolute value of its determinant is unity. Additionally, the determinant of **U** is nonnegative, and hence,

$$\mu = \det(\mathbf{U}) \tag{5.78b}$$

which shows that the manipulability is the product of the eigenvalues of **U** or equivalently, of the singular values of **J**. Now, the product of those singular values, in the geometric interpretation of the mapping induced by **J**, is proportional to the volume of the ellipsoid at hand, and hence, μ can be interpreted as a measure of the volume of that ellipsoid. It is apparent that the manipulability defined in Eq. (5.78b) is posture-dependent. For example, if **J** is singular, at least one of the semiaxes of the ellipsoid vanishes, and so does its volume. Manipulators at singular configurations thus have a manipulability of zero.

Now, if we want to use the concept of manipulability to define a posture-independent kinetostatic index, we have to define this index in a *global sense*. This can be done in the same way as the magnitude of a vector is defined, namely, as the sum of the squares of its components. In this way, the *global manipulability* can be defined as the integral of a certain power of the manipulability over the whole workspace of the manipulator, which would amount to defining the index as a *norm* of the manipulability in a space of functions.[5] For example, we can use the maximum manipulability attained over the whole workspace, thereby ending up with what would be like a *Chebyshev norm*[6]; alternatively, we can use the *root-mean square* (rms) value of the manipulability, thereby ending up with a measure similar to the Euclidean norm.

The condition number of a square matrix is a measure of the relative roundoff-error amplification of the computed results upon solving a linear system of equations associated with that matrix, with respect to the relative roundoff error of the data (Dahlquist and Björck 1974; Golub and Van Loan 1989). Based on the condition number of the Jacobian, a posture-independent *kinetostatic conditioning index* of robotic manipulators can be defined as a global measure of the condition number.

The definition of the condition number (Golub and Van Loan 1989) requires that all the entries of the matrix at hand bear the same physical units, which we assume first, in order to introduce the concept. The more frequent case of disparate units will be treated in the sequel. The condition number of a *dimensionally homogeneous* Jacobian $\mathbf{J}$ is defined as

$$\kappa(\mathbf{J}) = \|\mathbf{J}\|\|\mathbf{J}^{-1}\| \tag{5.79}$$

where $\|\cdot\|$ stands for a matrix norm (Golub and Van Loan 1989). While any norm can be used in the above definition, the one that is most convenient for our purposes is the *Frobenius norm* $\|\cdot\|_F$, defined as[7]

$$\|\mathbf{J}\|_F = \sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{J}\mathbf{J}^T)} = \sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{V}^2)} \tag{5.80a}$$

where we have assumed that $\mathbf{J}$ is of $n \times n$. Moreover, from the polar-decomposition theorem and Theorem 2.6.4, one can readily verify that

[5] Lack of familiarity with the mathematics of functions regarded as elements of vector spaces, what is called *functional analysis*, should not discourage the reader from continuing, for the balance of the book does not depend on these concepts.

[6] A norm is a generalization of the absolute value of real numbers, but applicable to arrays. In the same way that a vector norm is a measure of the "size" of the vector components, a matrix norm is a measure of the "size" of the matrix entries. In this vein, the Chebyshev norm of a given vector (matrix) is the largest absolute value of its components (entries).

[7] Actually, the definition of Eq. (5.80a) yields what is known as the *weighted* Frobenius norm, which gives a unit norm for the $n \times n$ identity matrix, *regardless* of the value of n.

$$\|\mathbf{J}\|_F = \sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{J}^T\mathbf{J})} = \sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{U}^2)} \tag{5.80b}$$

Now, since the trace of a matrix is nothing but the sum of the matrix eigenvalues, it is apparent that the Frobenius norm is identical with the rms value of the set of singular values of the given matrix. Likewise,

$$\|\mathbf{J}^{-1}\|_F = \sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{J}^{-1}\mathbf{J}^{-T})} = \sqrt{\frac{1}{n}\mathrm{tr}[(\mathbf{J}^T\mathbf{J})^{-1}]} = \sqrt{\frac{1}{n}\mathrm{tr}[(\mathbf{J}\mathbf{J}^T)^{-1}]} \tag{5.81}$$

and hence, computing the Frobenius norm of $\mathbf{J}^{-1}$ requires the inversion not of $\mathbf{J}$ itself, but rather that of $\mathbf{J}^T\mathbf{J}$, or of $\mathbf{J}\mathbf{J}^T$ for that matter. Furthermore, while $\mathbf{J}$ is not frame-invariant under a change of Cartesian-coordinate frame, $\mathbf{J}\mathbf{J}^T$ is. As a consequence, the latter lends itself better to a symbolic inversion than $\mathbf{J}$ itself. Hence, the *Frobenius condition number* κ_F is derived as

$$\kappa_F = \frac{1}{n}\sqrt{\mathrm{tr}(\mathbf{U}^2)\mathrm{tr}(\mathbf{U}^{-2})} = \frac{1}{n}\sqrt{\mathrm{tr}(\mathbf{V}^2)\mathrm{tr}(\mathbf{V}^{-2})} \tag{5.82}$$

Furthermore, if the *matrix 2-norm* is used in definition (5.79), then

$$\|\mathbf{J}\|_2 = \max_i\{\sigma_i\} \equiv \sigma_M, \quad \|\mathbf{J}^{-1}\|_2 \equiv \max_i\left\{\frac{1}{\sigma_i}\right\} = \frac{1}{\sigma_m} \tag{5.83a}$$

where

$$\sigma_m \equiv \min_i\{\sigma_i\} \tag{5.83b}$$

It is noteworthy that both the Frobenius norm and the 2-norm are given in terms of the matrix singular values. As a consequence, these two norms are frame-invariant. The *2-norm condition number* $\kappa_2(\mathbf{J})$ is thus given by

$$\kappa_2(\mathbf{J}) \equiv \frac{\sigma_M}{\sigma_m} \tag{5.84}$$

Now we can state a fundamental result:

Theorem 5.8.1. *The condition number based on either the 2-norm or the Frobenius norm of the robot Jacobian is invariant to changes of frame. In this light, the said condition numbers are immutable under a change of b_1, which amounts to a translation of frame $\mathcal{F}_1$, or of θ_1, which amounts to looking at the robot from a frame rotated by this angle about Z_1. Moreover, angle α_n not depending on the robot architecture, but on the location of the task frame, neither influences the same condition numbers.*

Note that, regardless of the norm adopted, the condition number can attain values from unity to infinity. Apparently, the condition number attains its minimum value of unity for matrices with identical singular values; such matrices map the unit ball into another ball, although of a different size, and are, thus, called *isotropic*. By extension, *isotropic manipulators* are those whose Jacobian matrix can attain isotropic values. On the other side of the spectrum, singular matrices have a smallest singular value that vanishes, and hence, their condition number is infinite. The condition number of $\mathbf{J}$ can be thought of as indicating the *distortion* of the unit ball in the space of joint-variables. The larger this distortion, the greater the condition number, the worst-conditioned Jacobians being those that are singular. For these, one of the semiaxes of the ellipsoid vanishes and the ellipsoid degenerates into what would amount to an elliptical disk in the three-dimensional space.

Now, if the entries of $\mathbf{J}$ have different units, its condition number is undefined, for we would face a problem of either adding or ordering from largest to smallest singular values of different units. Staffetti et al. (2002) called kinetostatic performance indices of manipulators with such a Jacobian matrix "non-invariant" to changes of norms. The same authors went on to claim that, because of this feature, such performance indices—Staffetti et al. refer to these indices as "manipulability indices"—-are not natural. We will leave aside the discussion of whether the indices at stake are invariant or not, to focus instead on means to cope with the problem at hand. We resolve the inconsistency of physical units by defining a *characteristic length*, by which we divide the Jacobian entries that have units of length, thereby producing a new Jacobian that is dimensionally homogeneous. We shall therefore divide our study into (a) manipulators for only positioning tasks, (b) manipulators for only orientation tasks, and (c) manipulators for both positioning and orientation tasks. The characteristic length will be introduced when studying the third category.

In the sequel, we will need an important property of isotropic matrices that is recalled below. First note that if $\mathbf{A}$ is isotropic, all its singular values are identical, say equal to σ, and hence, matrices $\mathbf{U}$ and $\mathbf{V}$ are proportional to the $n \times n$ identity matrix, i.e.,

$$\mathbf{U} = \mathbf{V} = \sigma \mathbf{1} \tag{5.85}$$

In this case, then,

$$\mathbf{A} = \sigma \mathbf{R} \tag{5.86a}$$

which means that isotropic square matrices are proportional to orthogonal matrices. As a consequence, then,

$$\mathbf{A}^T \mathbf{A} = \mathbf{A}\mathbf{A}^T = \sigma^2 \mathbf{1} \tag{5.86b}$$

Given an arbitrary manipulator of the serial type with a Jacobian matrix whose entries all have the same units, we can calculate its condition number and use a global measure of this to define a posture-independent kinetostatic index. Let

κ_m be the minimum value attained by the condition number of the dimensionally homogeneous Jacobian over the whole workspace, regardless of the norm adopted. Note that $1/\kappa_m$ can be regarded as a Chebyshev norm[8] of the reciprocal of the condition number, because now $1/\kappa_m$ represents the maximum value of this reciprocal in the whole workspace. We then introduce a posture-independent performance index, the *kinetostatic conditioning index*, or KCI for brevity, defined as

$$\text{KCI} = \frac{1}{\kappa_m} \times 100\% \tag{5.87}$$

Notice that since the condition number is bounded from below, the KCI is bounded from above by a value of 100%. Manipulators with a KCI of 100% are those identified above as isotropic.

5.8.1 Positioning Manipulators

Here, again, we shall distinguish between planar and spatial manipulators. These are studied separately.

Planar Manipulators

If the manipulator of Fig. 5.6 is limited to positioning tasks, we can dispense with its third axis, the manipulator thus reducing to the one shown in Fig. 5.7; its Jacobian reduces correspondingly to

$$\mathbf{J} = \begin{bmatrix} \mathbf{E}\mathbf{s}_1 & \mathbf{E}\mathbf{s}_2 \end{bmatrix}$$

with $\mathbf{s}_i$ denoting the two-dimensional versions of vectors $\mathbf{r}_i$ of the Denavit–Hartenberg notation, as introduced in Fig. 5.1. Now, if we want to design this manipulator for maximum manipulability, we resort to Eq. (5.78a), whence $\mu = |\det(\mathbf{J})|$. First, notice that

$$\det(\mathbf{J}) = \det(\mathbf{E}\begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{bmatrix}) = \det(\mathbf{E})\det(\begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{bmatrix})$$

and since matrix $\mathbf{E}$ is orthogonal, its determinant equals unity. Thus, the determinant of interest is now calculated using Fact 5.7.3 of Sect. 5.7, namely,

$$\det(\mathbf{J}) = -\mathbf{s}_1^T \mathbf{E}\mathbf{s}_2 \tag{5.88}$$

[8] In a space of functions.

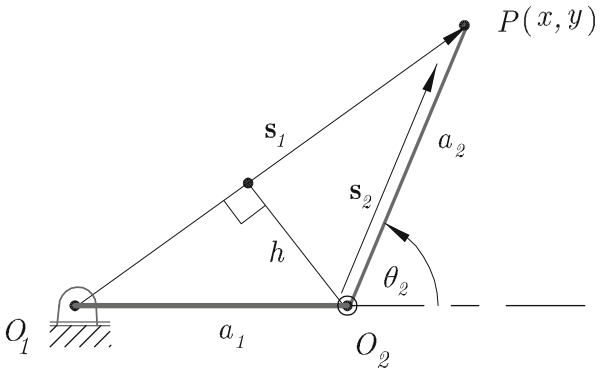


Fig. 5.10 A planar, two-axis positioning manipulator, with $\theta_1 = 0$

Therefore,

$$\mu = |\mathbf{s}_1^T \mathbf{E} \mathbf{s}_2| \equiv \|\mathbf{s}_1\| \|\mathbf{s}_2\| |\sin(\mathbf{s}_1, \mathbf{s}_2)|$$

where $(\mathbf{s}_1, \mathbf{s}_2)$ stands for the angle between the two vectors inside the parentheses. Now let us denote the manipulator reach with R, i.e., $R = a_1 + a_2$, and let $a_k = R\rho_k$, where ρ_k, for $k = 1, 2$, is a dimensionless number. As illustrated in Fig. 5.10, $\|\mathbf{s}_2\| \, |\sin(\mathbf{s}_1, \mathbf{s}_2)| = h$, the height of triangle $O_1 O_2 P$ of base $\overline{O_1 P}$, and hence, μ turns out to be twice the area of the same triangle, with the notation adopted at the outset.

Moreover, in terms of the base $\overline{O_1 O_2} = a_1$ and the height $a_2 |\sin \theta_2|$, the area of the triangle becomes $a_1 a_2 |\sin \theta_2|/2$, and hence,

$$\mu = a_1 a_2 |\sin \theta_2| = R^2 \rho_1 \rho_2 |\sin \theta_2| \tag{5.89a}$$

with ρ_1 and ρ_2 subject to

$$\rho_1 + \rho_2 = 1 \tag{5.89b}$$

The design task at hand, i.e., finding a_1 and a_2, can then be formulated as an optimization problem aimed at maximizing μ as given in Eq. (5.89a) over ρ_1 and ρ_2, subject to the constraint (5.89b). This optimization problem can be readily solved using, for example, Lagrange multipliers, thereby obtaining

$$\rho_1 = \rho_2 = \frac{1}{2}, \quad \theta_2 = \pm \frac{\pi}{2}$$

the absolute value of $\sin \theta_2$ attaining its maximum value when $\theta_2 = \pm 90°$. The maximum manipulability thus becomes

$$\mu_{\max} = \frac{R^2}{4} \tag{5.90}$$

Incidentally, the equal-length condition maximizes the workspace volume as well.

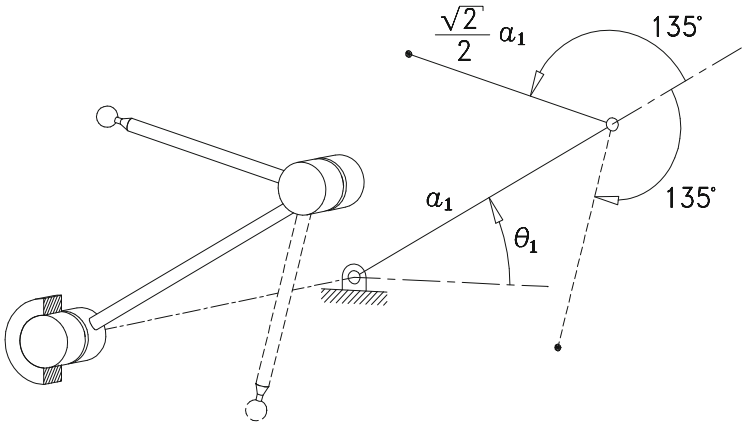


Fig. 5.11 A two-axis isotropic manipulator

On the other hand, if we want to minimize the condition number of $\mathbf{J}$, we should aim at rendering $\mathbf{J}$ isotropic, which means that the product $\mathbf{J}^T\mathbf{J}$ should be proportional to the identity matrix, and so,

$$\begin{bmatrix} \mathbf{s}_1^T\mathbf{s}_1 & \mathbf{s}_1^T\mathbf{s}_2 \\ \mathbf{s}_1^T\mathbf{s}_2 & \mathbf{s}_2^T\mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

where σ is the repeated singular value of $\mathbf{J}$. Hence, for $\mathbf{J}$ to be isotropic, all we need is that the two vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ have the same norm and that they lie at right angles. The solution is a manipulator with link lengths observing a ratio of $\sqrt{2}/2$, i.e., with $a_2/a_1 = \sqrt{2}/2$, and the two link axes at an angle of 135°, as depicted in Fig. 5.11. Manipulators of the above type, used as mechanical fingers, were investigated by Salisbury and Craig (1982), who found that these manipulators can be rendered isotropic if given the foregoing dimensions and configured as shown in Fig. 5.11.

Spatial Manipulators

Now we have a manipulator like that depicted in Fig. 4.11, its Jacobian matrix taking on the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \mathbf{e}_3 \times \mathbf{r}_3 \end{bmatrix} \tag{5.91}$$

The condition for isotropy of this kind of manipulator takes on the form of Eq. (5.86b), which thus leads to

$$\begin{bmatrix} \|\mathbf{e}_1 \times \mathbf{r}_1\|^2 & (\mathbf{e}_1 \times \mathbf{r}_1)^T(\mathbf{e}_2 \times \mathbf{r}_2) & (\mathbf{e}_1 \times \mathbf{r}_1)^T(\mathbf{e}_3 \times \mathbf{r}_3) \\ (\mathbf{e}_1 \times \mathbf{r}_1)^T(\mathbf{e}_2 \times \mathbf{r}_2) & \|\mathbf{e}_2 \times \mathbf{r}_2\|^2 & (\mathbf{e}_2 \times \mathbf{r}_2)^T(\mathbf{e}_3 \times \mathbf{r}_3) \\ (\mathbf{e}_1 \times \mathbf{r}_1)^T(\mathbf{e}_3 \times \mathbf{r}_3) & (\mathbf{e}_2 \times \mathbf{r}_2)^T(\mathbf{e}_3 \times \mathbf{r}_3) & \|\mathbf{e}_3 \times \mathbf{r}_3\|^2 \end{bmatrix} = \sigma^2\mathbf{1} \tag{5.92}$$

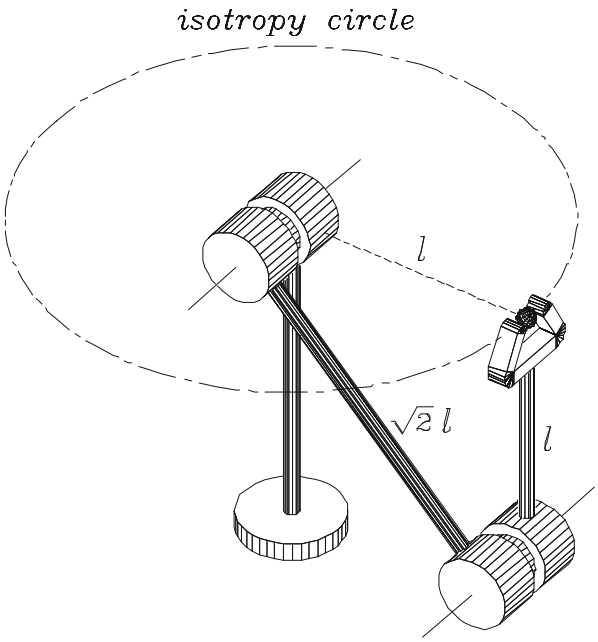


Fig. 5.12 An isotropic manipulator for three-dimensional positioning tasks

Hence, the manipulator under study can be postured so as to attain isotropy if its dimensions are chosen so that its three columns have the same Euclidean norm and are mutually orthogonal. These conditions can be attained by various designs, one example being the manipulator of Fig. 4.17. Another isotropic manipulator for three-dimensional positioning tasks is displayed in Fig. 5.12.

Note that the manipulator of Fig. 5.12 has an orthogonal architecture, the ratio of its last link length to the length of the intermediate link being, as in the two-dimensional case, $\sqrt{2}/2$. Since the first axis does not affect singularities, neither does it affect isotropy, and hence, not only does one location of the operation point exist that renders the manipulator isotropic, but a whole locus, namely, the circle known as the *isotropy circle*, indicated in the same figure. By the same token, the manipulator of Fig. 5.11 has an isotropy circle centered at the center of the first joint, with a radius of $(\sqrt{2}/2)a_1$.

5.8.2 Orienting Manipulators

We now have a three-revolute manipulator like that depicted in Fig. 4.19, its Jacobian taking on the simple form

$$\mathbf{J} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \tag{5.93}$$

and hence, the first isotropy condition of eq. (5.86b) leads to

$$\mathbf{J}^T\mathbf{J} = \begin{bmatrix} \mathbf{e}_1^T\mathbf{e}_1 & \mathbf{e}_1^T\mathbf{e}_2 & \mathbf{e}_1^T\mathbf{e}_3 \\ \mathbf{e}_2^T\mathbf{e}_2 & \mathbf{e}_2^T\mathbf{e}_2 & \mathbf{e}_2^T\mathbf{e}_3 \\ \mathbf{e}_3^T\mathbf{e}_1 & \mathbf{e}_3^T\mathbf{e}_2 & \mathbf{e}_3^T\mathbf{e}_3 \end{bmatrix} = \sigma^2\mathbf{1} \tag{5.94}$$

What the foregoing condition states is that a spherical wrist for orienting tasks is isotropic if its three unit vectors $\{\mathbf{e}_k\}_1^3$ are so laid out that they are mutually orthogonal, which thus yields $\mathbf{J} = \mathbf{1}$, the 3×3 identity matrix. Since the three singular values of $\mathbf{1}$ are all equal to unity, i.e., $\sigma = 1$, $\mathbf{J}^T\mathbf{J} = \mathbf{J}\mathbf{J}^T = \mathbf{1}$ as well. This is the case in orthogonal wrists when the two planes defined by the corresponding pairs of neighboring axes are at right angles. In summary, then, orthogonal wrists, which are rather frequent among industrial manipulators, are isotropic. Here we have an example of engineering insight leading to an optimum design, for such wrists existed long before isotropy was introduced as a manipulator design criterion. Moreover, notice that from the results of Sect. 4.4.2, spherical manipulators with an orthogonal architecture have a maximum workspace volume. That is, isotropic manipulators of the spherical type have two optimality properties: they have both a maximum workspace volume and a maximum KCI. Apparently, the manipulability of orthogonal spherical wrists is also optimal, as the reader is invited to verify, when the wrist is postured so that its three axes are mutually orthogonal. In this posture, the manipulability of the wrist is unity.

5.8.3 Positioning and Orienting Manipulators

We saw already in Sect. 5.8.1 that the optimization of the two indices studied here—the Jacobian condition number and manipulability—leads to different manipulators. In fact, the two indices entail even deeper differences, as we shall see presently. First and foremost, as we shall prove for both planar and spatial manipulators, the manipulability μ is independent of the operation point P of the end-effector, while the condition number is not. One more fundamental difference is that while calculating the manipulability of manipulators intended for both positioning and orienting tasks poses no problem, the condition number cannot be calculated, at least directly, for this kind of manipulator. Indeed, in order to determine the condition number of the Jacobian matrix, we must either add or order from largest to smallest its singular values. However, in the presence of positioning and orienting tasks, three of these singular values, those associated with orientation, are dimensionless, while those associated with positioning have units of length, thereby making impossible such an ordering. We resolve this dimensional inhomogeneity by introducing a normalizing *characteristic length*. Upon dividing the three *positioning* rows, i.e., the bottom rows, of the Jacobian by this length, a nondimensional Jacobian is obtained whose singular values are nondimensional as well. The characteristic length is

then defined as the normalizing length that renders the condition number of the Jacobian matrix a minimum. While this definition does not bear a direct geometric interpretation, in general, we shall see that such an interpretation is possible for isotropic manipulators. Below we shall determine the characteristic length for isotropic manipulators; determining the same for nonisotropic manipulators requires solving a minimization problem that calls for numerical techniques, as illustrated with examples.

Planar Manipulators

In the sequel, we will need the planar counterpart of the twist-transfer formula of Sect. 3.4.2. First, we denote by $\mathbf{t}_A$ the three-dimensional twist of a rigid body undergoing planar motion—introduced in Eq. (5.60)—when defined at a point A; when defined at point B, the corresponding twist is denoted by $\mathbf{t}_B$, i.e.,

$$\mathbf{t}_A \equiv \begin{bmatrix} \omega \\ \dot{\mathbf{a}} \end{bmatrix}, \quad \mathbf{t}_B \equiv \begin{bmatrix} \omega \\ \dot{\mathbf{b}} \end{bmatrix} \tag{5.95}$$

The relation between the two twists, or the *planar twist-transfer formula*, is given by a linear transformation $\mathbf{U}$ as

$$\mathbf{t}_B = \mathbf{U}\mathbf{t}_A \tag{5.96a}$$

where $\mathbf{U}$ is now defined as a 3×3 matrix, namely,

$$\mathbf{U} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{E}(\mathbf{b} - \mathbf{a}) & \mathbf{1}_2 \end{bmatrix} \tag{5.96b}$$

with $\mathbf{a}$ and $\mathbf{b}$ representing the position vectors of points A and B, and $\mathbf{1}_2$ standing for the 2×2 identity matrix. Moreover, $\mathbf{U}$ is, not surprisingly, a member of the 3×3 unimodular group, i.e.,

$$\det(\mathbf{U}) = 1$$

Because of the planar twist-transfer formula, the Jacobian defined at an operation point B is related to that defined at an operation point A of the same end-effector by the same linear transformation $\mathbf{U}$, i.e., if we denote the two Jacobians by $\mathbf{J}_A$ and $\mathbf{J}_B$, then

$$\mathbf{J}_B = \mathbf{U}\mathbf{J}_A \tag{5.97}$$

and if we denote by μ_A and μ_B the manipulability calculated at points A and B, respectively, then

$$\mu_B = |\det(\mathbf{J}_B)| = |\det(\mathbf{U})||\det(\mathbf{J}_A)| = |\det(\mathbf{J}_A)| = \mu_A \tag{5.98}$$

thereby proving that the manipulability is insensitive to a change of operation point, or to a change of end-effector, for that matter. Note that a similar analysis for the condition number cannot be completed at this stage because, as pointed out earlier, the condition number of these Jacobian matrices cannot even be calculated directly.

In order to resolve the foregoing dimensional inhomogeneity, we introduce the characteristic length L, which will be defined as that rendering the Jacobian dimensionally homogeneous and optimally conditioned, i.e., with a minimum condition number. We thus introduce the normalized Jacobian matrix as

$$\bar{\mathbf{J}} \equiv \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{L}\mathbf{E}\mathbf{s}_1 & \frac{1}{L}\mathbf{E}\mathbf{s}_2 & \frac{1}{L}\mathbf{E}\mathbf{s}_3 \end{bmatrix} \tag{5.99}$$

Now, if we want to size the manipulator at hand by properly choosing its geometric parameters so as to render it isotropic, we must observe the isotropy condition, e.g., the second of Eq. (5.86b), which readily leads to

$$\begin{bmatrix} 3 & \frac{1}{L}\left(\sum_1^3 \mathbf{s}_k^T\right)\mathbf{E}^T \\ \frac{1}{L}\mathbf{E}\sum_1^3 \mathbf{s}_k & \frac{1}{L^2}\mathbf{E}\left[\sum_1^3 \left(\mathbf{s}_k\mathbf{s}_k^T\right)\right]\mathbf{E}^T \end{bmatrix} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \tag{5.100}$$

and hence,

$$\sigma^2 = 3 \tag{5.101a}$$

$$\left(\sum_1^3 \mathbf{s}_k^T\right)\mathbf{E}^T \quad \text{or} \quad \sum_1^3 \mathbf{s}_k = \mathbf{0} \tag{5.101b}$$

$$\frac{1}{L^2}\mathbf{E}\left(\sum_1^3 (\mathbf{s}_k\mathbf{s}_k^T)\right)\mathbf{E}^T = \sigma^2\mathbf{1}_2 \tag{5.101c}$$

What Eq. (5.101a) states is simply that the triple singular value of the isotropic $\mathbf{J}$ is $\sqrt{3}$; Eq. (5.101b) states, in turn, that the operation point is the centroid of the centers of all manipulator joints if its Jacobian matrix is isotropic. Now, in order to gain insight into Eq. (5.101c), we note that since $\mathbf{E}$ is orthogonal and $\sigma^2 = 3$, this equation can be rewritten in a simpler form, namely,

$$\frac{1}{L^2}\left(\sum_1^3 (\mathbf{s}_k\mathbf{s}_k^T)\right) = (3)\mathbf{1}_2 \tag{5.102}$$

Further, if we recall the definition of the moment of inertia of a rigid body, we can immediately realize that the moment of inertia $\mathbf{I}_P$ of a set of particles of unit mass located at the centers of the manipulator joints, with respect to the operation point P, is given by

$$\mathbf{I}_P \equiv \sum_1^3 \left(\|\mathbf{s}_k\|^2\mathbf{1}_2 - \mathbf{s}_k\mathbf{s}_k^T\right) \tag{5.103}$$

from which it is apparent that the moment of inertia of the set comprises two parts, the first being isotropic—it is a multiple of the 2×2 identity matrix—the second not necessarily so. However, the second part has the form of the left-hand side of Eq. (5.102). Hence, Eq. (5.102) states that if the manipulator under study is isotropic, then its joint centers are located, at the isotropic configuration, at the corners of a triangle that has *circular inertial symmetry*. What we mean by this is that the 2×2 moment of inertia of the set of particles, with entries I_{xx}, I_{xy}, and I_{yy}, is similar to that of a circle, i.e., with $I_{xx} = I_{yy}$ and $I_{xy} = 0$. An obvious candidate is an equilateral triangle, the operation point thus coinciding with the center of the triangle. Since the corners of an equilateral triangle are at equal distances d from the center, and these distances are nothing but $\|\mathbf{s}_k\|$, the condition below is readily derived for isotropy:

$$\|\mathbf{s}_k\|^2 = d^2, \quad k = 1, 2, 3 \tag{5.104}$$

In order to compute the characteristic length of the manipulator under study, let us take the trace of both sides of Eq. (5.102), thereby obtaining

$$\frac{1}{L^2} \sum_{1}^{3} \|\mathbf{s}_k\|^2 = 6$$

and hence, upon substituting Eq. (5.104) into the foregoing relation, an expression for the characteristic length, *as pertaining to planar isotropic manipulators*, is readily derived:

$$L = \frac{\sqrt{2}}{2} d \tag{5.105}$$

It is now a simple matter to show that the three link lengths of this isotropic manipulator are $a_1 = a_2 = \sqrt{3}d$ and $a_3 = d$. Such a manipulator is sketched at an isotropic posture in Fig. 5.13.

We now can give a geometric interpretation of the characteristic length for the case at hand: To this end, we look at the manipulator of Fig. 5.13 from an arbitrary viewpoint outside of the manipulator plane, as depicted in Fig. 5.4. Let this plane be X–Y, with origin at O_1, and X-axis directed towards O_2. Next, we look at a point O on the normal to the X–Y plane passing through the operation point P, a distance h from P (Fig. 5.14).

Further, we define vectors $\{\mathbf{r}_i\}_1^3$ as

$$\mathbf{r}_i = \overrightarrow{OO}_i, \quad i = 1, 2, 3$$

Upon imposing the condition that the set $\{\mathbf{r}_i\}_1^3$ be orthogonal, we find h as

$$h = \frac{\sqrt{2}}{2} = L \tag{5.106}$$

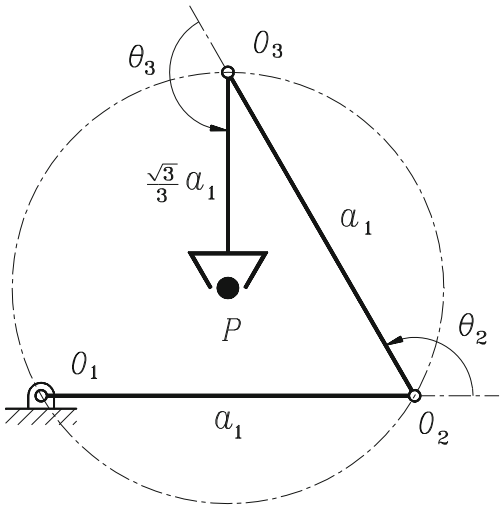


Fig. 5.13 The planar 3-R isotropic manipulator

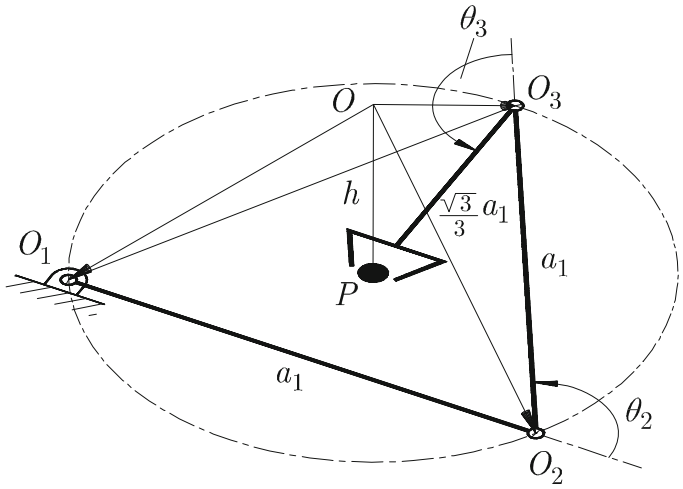


Fig. 5.14 A geometric interpretation of the characteristic length of the planar 3R isotropic manipulator

Therefore, the characteristic length L renders matrix $L\bar{\mathbf{J}}$ isotropic. In fact, this matrix becomes

$$L\bar{\mathbf{J}} = \begin{bmatrix} L & L & L \\ \mathbf{Es}_1 & \mathbf{Es}_2 & \mathbf{Es}_3 \end{bmatrix} \tag{5.107}$$

for $\{\mathbf{s}_i\}_1^3$ valued at the isotropic posture of Fig. 5.13. Notice that the difference between the Jacobian matrix defined in Eq. (5.60) and matrix $L\bar{\mathbf{J}}$ lies only in their first row. Obviously, the former is not dimensionally homogeneous; the latter is.

Spatial Manipulators

The entries of the Jacobian of a six-axis manipulator intended for both positioning and orienting tasks are dimensionally inhomogeneous as well. Indeed, as discussed in Sect. 5.2, the ith column of $\mathbf{J}$ is composed of the *Plücker coordinates* of the ith axis of the manipulator, namely,

$$\mathbf{J} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_5 & \mathbf{e}_6 \\ \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \mathbf{e}_3 \times \mathbf{r}_3 & \mathbf{e}_4 \times \mathbf{r}_4 & \mathbf{e}_5 \times \mathbf{r}_5 & \mathbf{e}_6 \times \mathbf{r}_6 \end{bmatrix} \tag{5.108}$$

Now it is apparent that the first three rows of $\mathbf{J}$ are dimensionless, whereas the remaining three, corresponding to the *moments* of the axes with respect to the operation point of the end-effector, have units of length. This dimensional inhomogeneity is resolved in the same way as in the case of planar manipulators for both positioning and orienting tasks, i.e., by means of a characteristic length. This length is defined, again, as the one that minimizes the condition number of the dimensionless Jacobian thus obtained. We then define the corresponding normalized Jacobian as

$$\bar{\mathbf{J}} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_5 & \mathbf{e}_6 \\ \frac{1}{L}\mathbf{e}_1 \times \mathbf{r}_1 & \frac{1}{L}\mathbf{e}_2 \times \mathbf{r}_2 & \frac{1}{L}\mathbf{e}_3 \times \mathbf{r}_3 & \frac{1}{L}\mathbf{e}_4 \times \mathbf{r}_4 & \frac{1}{L}\mathbf{e}_5 \times \mathbf{r}_5 & \frac{1}{L}\mathbf{e}_6 \times \mathbf{r}_6 \end{bmatrix} \tag{5.109}$$

and hence, the second isotropy condition of Eq. (5.86b) leads to

$$\sum_1^6 \mathbf{e}_k \mathbf{e}_k^T = \sigma^2 \mathbf{1} \tag{5.110a}$$

$$\sum_1^6 \mathbf{e}_k (\mathbf{e}_k \times \mathbf{r}_k)^T = \mathbf{O} \tag{5.110b}$$

$$\frac{1}{L^2} \sum_1^6 (\mathbf{e}_k \times \mathbf{r}_k)(\mathbf{e}_k \times \mathbf{r}_k)^T = \sigma^2 \mathbf{1} \tag{5.110c}$$

where $\mathbf{1}$ is the 3×3 identity matrix, and $\mathbf{O}$ is the 3×3 zero matrix. Now, if we take the trace of both sides of Eq. (5.110a), we obtain

$$\sigma^2 = 2 \quad \text{or} \quad \sigma = \sqrt{2}$$

Furthermore, we take the trace of both sides of Eq. (5.110c), which yields

$$\frac{1}{L^2} \sum_1^6 \|\mathbf{e}_k \times \mathbf{r}_k\|^2 = 3\sigma^2$$

But $\|\mathbf{e}_k \times \mathbf{r}_k\|^2$ is nothing but the square of the distance d_k of the kth revolute axis to the operation point, the foregoing equation thus yielding

$$L = \sqrt{\frac{1}{6}\sum_{1}^{6} d_k^2}$$

i.e., *the characteristic length of a spatial six-revolute isotropic manipulator is the root-mean square of the distances of the revolute axes to the operation point when the robot finds itself at the posture of minimum condition number.*

Furthermore, Eq. (5.110a) states that if $\{\,\mathbf{e}_k\,\}_1^6$ is regarded as the set of position vectors of points $\{\,P_k\,\}_1^6$ on the surface of the unit sphere, then the moment-of-inertia matrix of the set of equal masses located at these points has *spherical symmetry*. What the latter means is that any direction of the three-dimensional space is a principal axis of inertia of the foregoing set. Likewise, Eq. (5.110c) states that if $\{\,\mathbf{e}_k \times \mathbf{r}_k\,\}_1^6$ is regarded as the set of position vectors of points $\{\,Q_k\,\}$ in the three-dimensional Euclidean space, then the moment-of-inertia matrix of the set of equal masses located at these points has spherical symmetry as well.

Now, in order to gain insight into Eq. (5.110b), let us take the axial vector of both sides of that equation, thus obtaining

$$\sum_{1}^{6} \mathbf{e}_k \times (\mathbf{e}_k \times \mathbf{r}_k) = \mathbf{0} \tag{5.111}$$

with $\mathbf{0}$ denoting the three-dimensional zero vector. Furthermore, let us denote by $\mathbf{E}_k$ the cross-product matrix of $\mathbf{e}_k$, the foregoing equation thus taking the form

$$\sum_{1}^{6} \mathbf{E}_k^2 \mathbf{r}_k = \mathbf{0}$$

However,

$$\mathbf{E}_k^2 = -\mathbf{1} + \mathbf{e}_k \mathbf{e}_k^T$$

for every k, and hence, Eq. (5.111) leads to

$$\sum_{1}^{6} (\mathbf{1} - \mathbf{e}_k \mathbf{e}_k^T)\mathbf{r}_k = \mathbf{0}$$

Moreover, $(\mathbf{1} - \mathbf{e}_k \mathbf{e}_k^T)\mathbf{r}_k$ is nothing but the normal component of $\mathbf{r}_k$ with respect to $\mathbf{e}_k$, as defined in Sect. 2.2. Let us denote this component by $\mathbf{r}_k^{\perp}$, thereby obtaining an alternative expression for the foregoing equation, namely,

$$\sum_{1}^{6} \mathbf{r}_k^{\perp} = \mathbf{0} \tag{5.112}$$

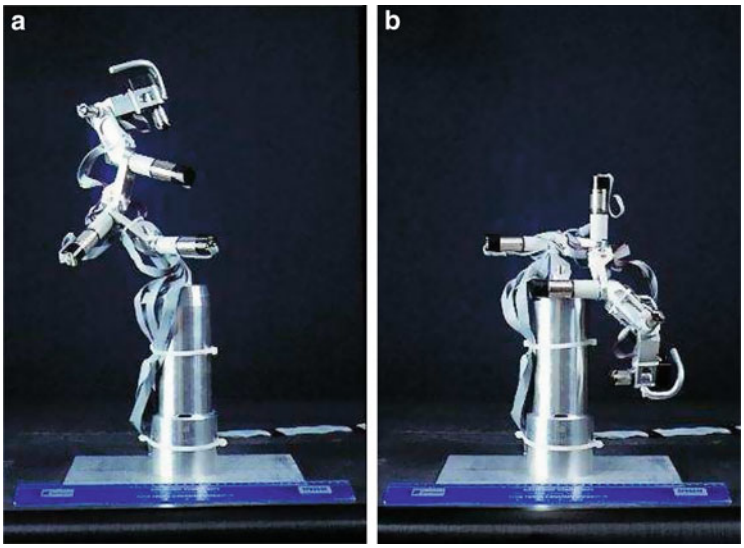


Fig. 5.15 DIESTRO, a six-axis isotropic manipulator in two postures: (**a**) with the arm extended upwards; (**b**) with the arm down, showing the orthogonality of the neighboring axes

Table 5.1 DH parameters of DIESTRO

i	a_i (mm)	b_i (mm)	α_i	θ_i
1	50	50	90°	θ_1
2	50	50	−90°	θ_2
3	50	50	90°	θ_3
4	50	50	−90°	θ_4
5	50	50	90°	θ_5
6	50	50	−90°	θ_6

The geometric interpretation of the above equation is now apparent: Let O'_k be the foot of the perpendicular to the kth revolute axis from the operation point P; then, $\mathbf{r}_k$ is the vector directed from O'_k to P. Therefore, *the operation point of an isotropic manipulator, configured at the isotropic posture, is the centroid of the set* $\{ O'_k \}_1^6$ *of perpendicular feet from the operation point*.

An axis layout that satisfies the foregoing isotropy conditions is obtained when the six axes are laid out along the diagonals of the faces of a cube. If the side of the cube has a length a, then all distances d_k from the cube centroid to the axes are identical: $d_k = a/2$, for $k = 1, \ldots, 6$.

A six-axis manipulator designed with an isotropic architecture, DIESTRO, is displayed in Fig. 5.15. The Denavit–Hartenberg parameters of this manipulator are given in Table 5.1. DIESTRO is characterized by identical link lengths a and offsets identical with this common link length, besides twist angles of 90° between all pairs of neighboring axes. Not surprisingly, the characteristic length of this manipulator is a.

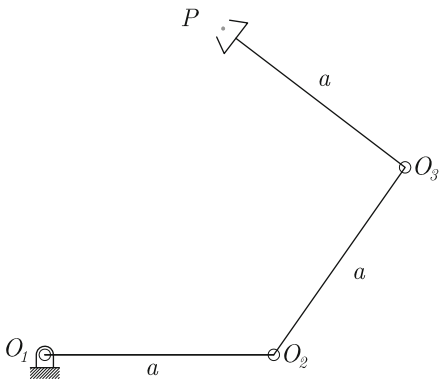


Fig. 5.16 Equilateral, three-revolute planar robot

5.8.4 *Computation of the Characteristic Length: Applications to Performance Evaluation*

We elaborate further on the concept of characteristic length. In order to provide a better grasp of the concept, we focus on its computation as pertaining to a given robot, that need not be isotropic. To do this, we include two examples, one planar and one spatial, industrial robot. Once a numerical value of the characteristic length is available, we can compute the minimum value of the condition number of the robot Jacobian, with which we can assess the robot kinetostatic performance by means of the KCI.

Example 5.8.1 (A Planar, Equilateral, Three-Revolute Robot). Compute the characteristic length of the robot of Fig. 5.16, depicting a posture in which θ_1 has been set equal to zero. What is the KCI of this robot?

Solution: We have $a_1 = a_2 = a_3 = a$ for the robot under study. In order to compute its length, we have to minimize the Jacobian condition number by a proper choice of the characteristic length L and the joint variables θ_2 and θ_3. We thus start by deriving an expression for the Jacobian:

$$\mathbf{J} = \begin{bmatrix} 1 & 1 & 1 \\ \mathbf{Es}_1 & \mathbf{Es}_2 & \mathbf{Es}_3 \end{bmatrix}$$

Now, we render $\mathbf{J}$ dimensionally homogeneous by introducing the characteristic length L, thus obtaining the *normalized Jacobian* $\bar{\mathbf{J}}$:

$$\bar{\mathbf{J}} = \begin{bmatrix} 1 & 1 & 1 \\ (1/L)\mathbf{Es}_1 & (1/L)\mathbf{Es}_2 & (1/L)\mathbf{Es}_3 \end{bmatrix}$$

From the manipulator geometry, we have,

$$\mathbf{Es}_1 = a\begin{bmatrix} -(s_1 + s_{12} + s_{123}) \\ c_1 + c_{12} + c_{123} \end{bmatrix}, \quad \mathbf{Es}_2 = a\begin{bmatrix} -(s_{12} + s_{123}) \\ c_{12} + c_{123} \end{bmatrix}, \quad \mathbf{Es}_3 = a\begin{bmatrix} -s_{123} \\ c_{123} \end{bmatrix}$$

with

$$c_1 \equiv \cos\theta_1, \quad c_2 \equiv \sin\theta_2, \quad c_{12} \equiv \cos(\theta_1 + \theta_2), \quad c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$$
$$s_1 \equiv \sin\theta_1, \quad s_2 \equiv \sin\theta_2, \quad s_{12} \equiv \sin(\theta_1 + \theta_2), \quad s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$$

Since we set $\theta_1 = 0$, because the first joint does not affect the condition number, the normalized Jacobian becomes

$$\bar{\mathbf{J}} = \begin{bmatrix} 1 & 1 & 1 \\ -r(s_2 + s_{23}) & -r(s_2 + s_{23}) & -rs_{23} \\ r(1 + c_2 + c_{23}) & r(c_2 + c_{23}) & rc_{23} \end{bmatrix}, \quad r \equiv \frac{a}{L}$$

the inverse of $\bar{\mathbf{J}}$, as derived with computer algebra, being

$$\bar{\mathbf{J}}^{-1} = \begin{bmatrix} s_3/s_2 & c_2/(rs_2) & 1/r \\ -(s_3 + s_{23})/s_2 & -(1 + c_2)/(rs_2) & -1/r \\ (s_2 + s_{23})/s_2 & 1/(rs_2) & 0 \end{bmatrix}$$

The square of the Frobenius-norm condition number of $\bar{\mathbf{J}}$ is now computed as

$$\kappa_F^2 = fg$$

with f and g defined as the square of the weighted Frobenius norms of $\bar{\mathbf{J}}$ and $\bar{\mathbf{J}}^{-1}$, respectively, i.e.,

$$f \equiv \|\bar{\mathbf{J}}\|_F^2 = \frac{1}{3}\mathrm{tr}(\bar{\mathbf{J}}^T\bar{\mathbf{J}}) = 1 + \left(2 + \frac{2}{3}c_2 + \frac{4}{3}c_3 + \frac{2}{3}c_{23}\right)r^2$$

and

$$g \equiv \|\bar{\mathbf{J}}^{-1}\|_F^2 = \frac{(s_2 + s_{23})^2 + (s_3 + s_{23})^2 + s_3^2}{3s_2^2} + \frac{2}{3}\frac{2 + c_2}{r^2 s_2^2}$$

which can be rewritten as

$$f = 1 + Ar^2$$

and

$$g = \frac{1}{3}\frac{D + E/r^2}{s_2^2}$$

with coefficients A, D and E independent of r, namely,

$$\begin{aligned} A &\equiv \frac{2}{3}(3 + c_2 + 2c_3 + c_{23}) \\ D &\equiv (s_2 + s_{23})^2 + (s_3 + s_{23})^2 + s_3^2 \\ E &\equiv 2(2 + c_2) \end{aligned}$$

We now have a classical minimization problem:

$$\kappa_F^2 \equiv fg \equiv \frac{1}{3}(1 + Ar^2)\frac{D + E/r^2}{s_2^2} \quad \rightarrow \min_{r,\theta_2,\theta_3}$$

where the characteristic length is implicit in r. While the foregoing problem is well posed, we should not forget that κ_F is unbounded from above. In order to gain better insight into the problem at hand, it is preferable to treat the problem as one of maximization of $1/\kappa_F$, or of its square, for that matter. As well, we can dispense with the constant factor $1/3$ in κ_F^2, which thus leads to the maximization problem below:

$$z \equiv \frac{Q}{P} \quad \rightarrow \max_{r,\theta_2,\theta_3}$$

with P and Q defined as

$$P \equiv ADr^4 + (AE + D)r^2 + E, \quad Q \equiv r^2 s_2^2$$

In order to obtain the optimum values of the three *design variables* r, θ_2, and θ_3, we need to set up the *normality conditions* of the problem at hand. These are readily obtained upon zeroing the gradient of κ_F^2 with respect to the vector of design variables, defined as $\mathbf{x} \equiv \begin{bmatrix} r & \theta_2 & \theta_3 \end{bmatrix}^T$. The said conditions are, thus,

$$\frac{\partial z}{\partial \mathbf{x}} \equiv \frac{\partial}{\partial \mathbf{x}}\left(\frac{Q}{P}\right) = \mathbf{0}_3$$

The three components of the above gradient, $\partial z/\partial r$, $\partial z/\partial \theta_2$, and $\partial z/\partial \theta_3$, are then derived using the general formula for the derivative of a rational expression:

$$\frac{\partial}{\partial x_i}\left(\frac{Q}{P}\right) = \frac{1}{P^2}\left(Q_i P - Q P_i\right)$$

where Q_i and P_i stand for $\partial Q/\partial x_i$ and $\partial P/\partial x_i$, with x_i taking values of r, θ_2, and θ_3, for $i = 1, 2, 3$, respectively. We thus have, using a similar notation for the partial derivatives of coefficients A, D and E:

$$
\begin{aligned}
P_1 &= 4ADr^3 + 2(AE + D)r, \quad Q_1 = 2rs_2^2 \\
P_2 &= (A_2D + AD_2)r^4 + (A_2E + AE_2 + D_2)r^2 + E_2, \quad Q_2 = 2r^2s_2c_2 \\
P_3 &= (A_3D + AD_3)r^4 + (A_3E + AE_3 + D_3)r^2 + E_3, \quad Q_3 = 0
\end{aligned}
$$

Apparently, $E_3 = 0$, the normality conditions thus simplifying to

$$
\begin{aligned}
\frac{\partial z}{\partial r} &\equiv -\frac{2rs_2^2}{P^2}\left(-ADr^4 + E\right) = 0 \\
\frac{\partial z}{\partial \theta_2} &\equiv \frac{r^2s_2^2}{P^2}\left\{[2ADc_2 - (A_2D + AD_2)s_2]r^4 + [2(AE + D)c_2\right. \\
&\qquad \left. -(A_2E + AE_2 + D_2)s_2]\,r^2 + 2Ec_2 - E_2s_2\right\} = 0 \\
\frac{\partial z}{\partial \theta_3} &\equiv -\frac{r^4s_2^2}{P^2}\left[(A_3D + AD_3)r^2 + (A_3E + D_3)\right] = 0
\end{aligned}
$$

thereby obtaining a system of three nonlinear equations in three unknowns, namely, the three design variables. Apparently, all three normality conditions are satisfied for either $r = 0$ or $s_2 = 0$, which just confirms that the normality conditions are sufficient for a point in the *design space* to be *stationary*; such a point can be a *local minimum*, a *local maximum* or a *saddle point*. The vanishing of the product rs_2 thus yields a minimum of z, which indicates $\kappa_F \to \infty$, $r = 0$ giving an *architecture singularity*, $s_2 = 0$ a posture singularity. We are not interested, for our purposes, on such a minimum, for which reason we assume henceforth that $rs_2 \neq 0$. Under this condition, the normality conditions thus yield the reduced system of equations

$$
\begin{aligned}
\phi_1 &\equiv -ADr^4 + E = 0 \\
\phi_2 &\equiv [2ADc_2 - (A_2D + AD_2)s_2]r^4 + [2(AE + D)c_2 - (A_2E + AE_2 + D_2)s_2]r^2 \\
&\qquad + 2Ec_2 - E_2s_2 = 0 \\
\phi_3 &\equiv (A_3D + AD_3)r^2 + (A_3E + D_3) = 0
\end{aligned}
$$

The problem at hand is thus solved by finding the roots of the foregoing system. We can do this by means of the Newton–Raphson method, for example, which (a) requires the 3×3 Jacobian matrix of the three foregoing equations, i.e., further differentiation, and (b) yields only one root, out of many, for one given *initial guess*, when the method converges at all. Moreover, given the *local* nature of the method, Newton–Raphson cannot tell whether one has found all possible roots of the system of equations.

An alternative, semigraphical approach, was introduced in Example 4.4.3. This approach consists in reducing the problem to finding the roots of two nonlinear equations in two unknowns; each equation, then, defines one contour in the plane of the two unknowns, the intersection points of the two contours yielding all possible real roots of the system at hand. In order to apply this approach to the above system, we have to eliminate one of the three unknowns from the system, the obvious candidate being r. We can do this by dialytic elimination, as introduced in Sect. 5.4.1. Given the special structure of the three given equations, it is simpler to eliminate r following a straightforward approach: First, we solve for r^4 from the first equation and for r^2 from the third, which yields:

$$r^4 = \frac{E}{AD}, \quad r^2 = -\frac{A_3E + D_3}{A_3D + AD_3}$$

Upon equating the above expression of r^4 with the square of its counterpart expression for r^2, we obtain

$$F(\theta_2, \theta_3) \equiv (AE - D)(AD_3^2 - A_3^2DE) = 0$$

Further, upon substituting the same expressions for r^2 and r^4 into equation $\phi_2 = 0$, we obtain

$$\begin{aligned} G(\theta_2, \theta_3) \equiv (AE - D)[&(AA_3DE_2 + A_2A_3DE - AD_2D_3)\sin\theta_2 \\ &-2AD(A_3E - D_3)\cos\theta_2] = 0 \end{aligned}$$

thereby obtaining a reduced system of two equations in two unknowns only, θ_2 and θ_3. The foregoing system admits further simplifications. Indeed, the common factor $AE - D$ turns out to be positive-definite, i.e., $AE - D > 0$ *for any values* of θ_2 and θ_3. While it is not obvious that the factor in question is positive-definite, its sign-definiteness was verified with the aid of computer algebra. To visualize this property, we include a three-dimensional rendering of the function as a surface in Fig. 5.17a and a side view of the same in Fig. 5.17b. Given that the factor in question is positive-definite, we can safely divide both sides of the two foregoing equations by this factor, which thus leads to two nonlinear equations in θ_2 and θ_3 defining contours $\mathcal{C}_1$ and $\mathcal{C}_2$ in the θ_2–θ_3 plane, namely,

$$\begin{aligned} \mathcal{C}_1: &\quad AD_3^2 - A_3^2DE = 0 \\ \mathcal{C}_2: &\quad (AA_3DE_2 + A_2A_3DE - AD_2D_3)\sin\theta_2 - 2AD(A_3E - D_3)\cos\theta_2 = 0 \end{aligned}$$

The two above contours are plotted in the θ_2–θ_3 plane in Fig. 5.18.

Apparently, to any optimum posture with joint center O_3 above line O_1O_2 corresponds a symmetrically located posture of the robot with O_3 below the above line. This means that all solutions (θ_2, θ_3) expected should be symmetric about the origin of the θ_2–θ_3 plane, which they are, as illustrated in Fig. 5.18a. That is, if a

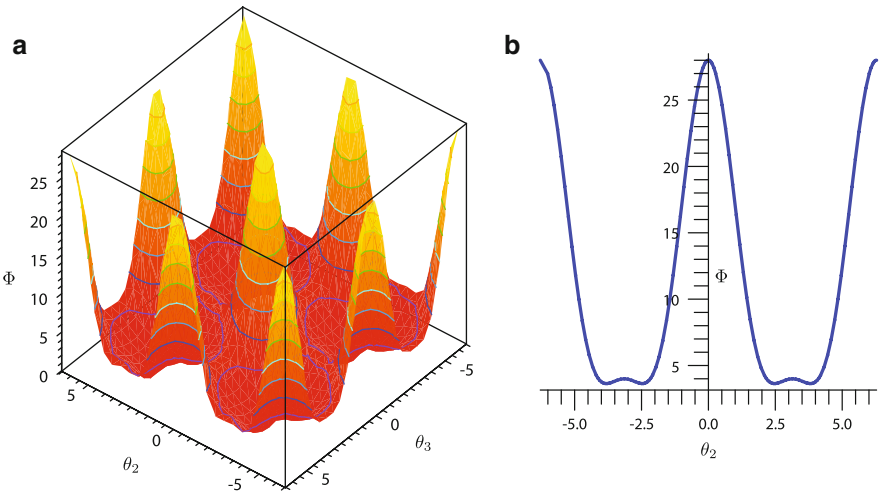


Fig. 5.17 A 3D rendering of the function $\Phi = AE - D$ vs. θ_2 and θ_3: (**a**) an isometric view; and (**b**) a view in the θ_2–Φ plane

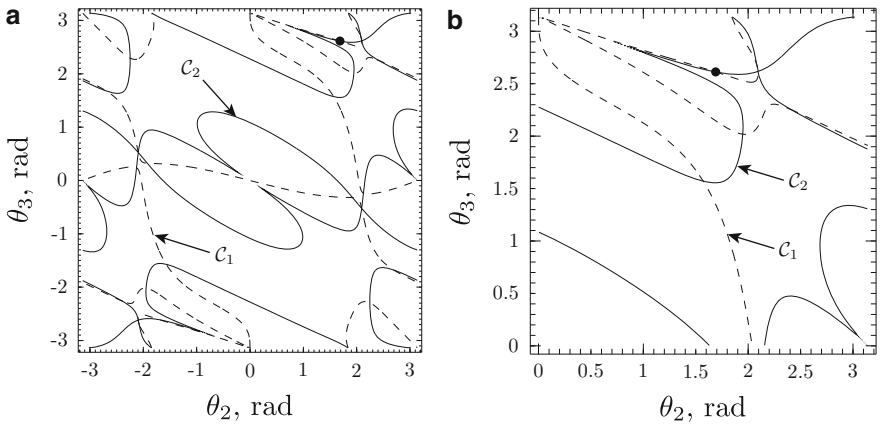


Fig. 5.18 Contours $\mathcal{C}_1$ and $\mathcal{C}_2$ in the θ_2–θ_3 plane: (**a**) in the $-\pi \leq \theta_2 \leq \pi$, $-\pi \leq \theta_3 \leq \pi$ range; (**b**) a zoom-in in the $0 \leq \theta_2 \leq \pi$, $0 \leq \theta_3 \leq \pi$ range

pair of numerical values $(\theta_2,\ \theta_3)$ verifies the normality conditions, then so does the pair $(-\theta_2,\ -\theta_3)$. By the same token, if we set $\theta_1 = \pi$ in the Jacobian expression, a similar set of normality conditions should be obtained, with the corresponding symmetry.

In light of the symmetry of the plots of Fig. 5.18a, we can focus on the first quadrant of the θ_2–θ_3 plane, and plot the zoom-in of Fig. 5.18a, showing only this quadrant. Moreover, it is apparent that contour $\mathcal{C}_1$ exhibits two double points, one at (0.9, 2.9), the other at (2.1, 2.6). Double points are likely to produce *spurious*

solutions[9]; hence, we discard those two double points, thereby leaving only five intersections of interest. As it turns out, the intersection detected by inspection at, roughly, $\theta_2 = 1.69$ rad, $\theta_3 = 2.61$ rad produces a maximum of $1/\kappa_F$. These rough values of the design variables were then refined using the Newton–Raphson method, with the foregoing rough values as initial guess.[10] The Newton–Raphson method, as implemented with Matlab code, yielded the refined solution displayed below in eight iterations:

$$\begin{aligned} \theta_2 &= 1.68910726900188 \text{ rad} = 96.77871763^\circ, \\ \theta_3 &= 2.61287852677543 \text{ rad} = 149.7069120^\circ, \\ r &= 2.040896177 \quad \Rightarrow \quad L = a/r = 0.4899808287 \text{ m} \end{aligned}$$

where we have recalled that a was specified as 1 m. The normalized Jacobian $\bar{\mathbf{J}}$ at the optimum posture is, moreover,

$$\bar{\mathbf{J}} = \begin{bmatrix} 1 & 1 & 1 \\ -0.1552 & -0.1552 & 1.8715 \\ 0.9858 & -1.0552 & -0.8143 \end{bmatrix}$$

with $\kappa_F = 1.1832$. Hence,

$$\text{KCI} = 84.52\%$$

Note that, if we use the 2-norm to define the condition number, i.e., if we minimize

$$\kappa_2(r, \theta_2, \theta_3) = \frac{\sigma_M}{\sigma_m}$$

then we cannot find expressions for the gradient of $1/\kappa_2^2$ with respect to the design variables because the objective function now is not *analytic*[11] in the design variables. That is, unlike the minimization of κ_F, now we do not have normality conditions. Nevertheless, κ_2 can still be minimized using a *direct-search method*, i.e., an

[9] A spurious solution is a set of numerical values of the roots of a system of equations that, although computed from a sound elimination procedure, does not verify the equations. Example 9.7.3 includes a case of a double point in a contour that yields spurious solutions.

[10] It is well known (Dahlquist and Björck 1974) that, close to a root, the Newton–Raphson method converges *quadratically*, i.e., the approximation to the root gains, roughly, two digits of accuracy at each iteration. Hence, the Newton–Raphson procedure will likely converge to the root closest to the given estimate.

[11] A real-valued function of a real argument is said to be analytic at one value of its argument if the function admits a series expansion at this value. To be true, the exception to the rule is the case of 2×2 matrices, which have an analytic κ_2.

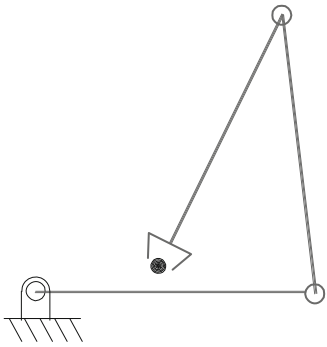


Fig. 5.19 Optimum configuration for a minimum κ_F

optimization method not relying on gradients, but only on objective-function evaluation. The objective function is that whose minimum, or maximum for that matter, is searched. In our case, the objective function to minimize is κ_2. Direct-search minimization methods are available in scientific software. Matlab, for example, uses the *Nelder–Mead simplex method*, as implemented in its `fminsearch` function. A local minimum value of κ_2 was found by `fminsearch` with the initial guess

$$r = 1.0, \quad \theta_2 = 6.0^\circ, \quad \theta_3 = 18.0^\circ$$

after 148 iterations, as

$$\kappa_2 = 1.9070$$

This value is attained at the values of the design variables given below:

$$r = 2.1650, \quad \theta_2 = 98.9785^\circ, \quad \theta_3 = 145.193^\circ$$

which yield $L = 0.4619$ m and a nondimensional Jacobian

$$\overline{\mathbf{J}} = \begin{bmatrix} 1 & 1 & 1 \\ -0.1898 & -0.1898 & 1.9488 \\ 0.8839 & -1.2813 & -0.9433 \end{bmatrix}$$

Shown in Fig. 5.19 is the given manipulator at the optimum posture under the condition number calculated using the Frobenius norm, the posture corresponding to the minimum condition number based on the 2-norm being indistinguishable from this one.

Example 5.8.2. Find the KCI and the characteristic length of the Fanuc Robot Arc Mate S series manipulator of 1990, whose DH parameters are given in Table 5.2.

Solution: We need the minimum value that the condition number κ_F of the normalized robot Jacobian can attain, in order to calculate its KCI as indicated

Table 5.2 DH parameters of the Fanuc Arc Mate S series manipulator of 1990

i	a_i (mm)	b_i (mm)	α_i	θ_i
1	200	810	90°	θ_1
2	600	0	0°	θ_2
3	130	−30	90°	θ_3
4	0	550	90°	θ_4
5	0	100	90°	θ_5
6	0	100	0°	θ_6

in Eq. (5.87). Now, the Fanuc Robot Arc Mate S series of 1990 is a six-revolute manipulator for positioning and orienting tasks. Hence, its Jacobian matrix has to be first recast in nondimensional form, as in Eq. (5.109). Next, we find L along with the joint variables that determine the posture of minimum condition number via an optimization procedure. Prior to the formulation of the underlying optimization problem, however, we recall Theorem 5.8.1, under which the first joint, accounting for motions of the manipulator as a single rigid body, does not affect its Jacobian condition number. By the same token, we align axes Z_6 and Z_7. As a consequence, θ_6 does not affect the Jacobian condition number either. We thus define the *design vector* $\mathbf{x}$ of the optimization problem at hand as a five-dimensional array, namely,

$$\mathbf{x} \equiv \begin{bmatrix} \theta_2 & \theta_3 & \theta_4 & \theta_5 & L \end{bmatrix}^T$$

and set up the optimization problem as

$$\min_{\mathbf{x}} \kappa_F(\mathbf{J})$$

Now, given the architecture of the robot at hand, a symbolic expression for $\mathbf{J}^{-1}$, or its dimensionless counterpart $\overline{\mathbf{J}}^{-1}$, not to speak of κ_F itself, is elusive, and hence, an approach like that of Example 5.8.1 appears rather unfeasible. We thus resort to a direct-search—as opposed to a gradient-based—procedure to solve the foregoing optimization problem. There are various methods of this kind at our disposal; the one we chose is, again, the Nelder–Mead *simplex method*, as implemented in Matlab within the `fminsearch` function, which was provided with the initial guess

$$\mathbf{x}_{\text{init}} = \begin{bmatrix} 26° & -56° & 195° & 107° & 341.738 \end{bmatrix}^T$$

The results reported by Matlab are displayed below:

$$\mathbf{x}_{\text{opt}} = \begin{bmatrix} 22.60° & -51.13° & -159.93° & 88° & 351.23 \end{bmatrix}^T$$

whose last entry, the characteristic length of the robot, is in mm, i.e.,

$$L = 351.23 \text{ mm}$$

Furthermore, the minimum condition number attained at the foregoing posture, with the characteristic length found above, is

$$\kappa_F = 1.2717$$

Therefore, the KCI of the Fanuc Robot Arc Mate S series manipulator of 1990 is

$$\text{KCI} = 78.63\%$$

To be sure, the KCI of this manipulator can still be improved dramatically by noting that the condition number is highly dependent on the location of the operation point of the end-effector. The robot DH parameters given in Table 5.2 do not account for the geometry of the EE.

5.9 Exercises

N.B.: Exercises 5.14–5.18 pertain to Sect. 5.8. They are thus to be assigned only if this section was covered in class.

5.1 Shown in Fig. 5.20 is a computer-generated model of DIESTRO, the robot displayed in Fig. 5.15, with a slightly modified EE. The Denavit–Hartenberg parameters of this robot are given in Table 5.3. Find the Jacobian matrix of the manipulator at the above configuration.

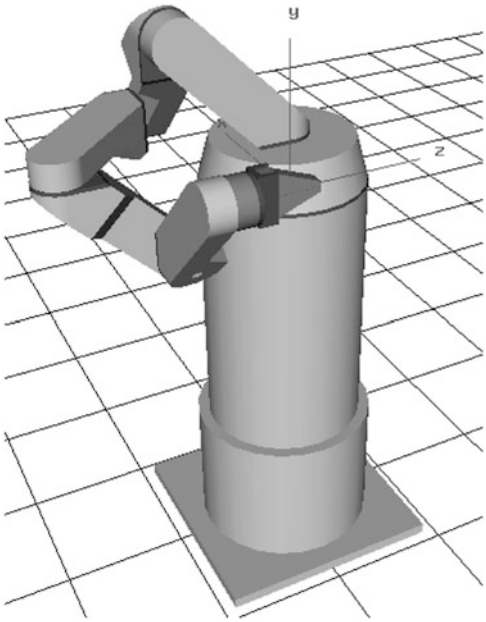


Fig. 5.20 A six-revolute manipulator

Table 5.3 DH parameters of the modified DIESTRO

i	a_i (mm)	b_i (mm)	α_i	θ_i
1	50	50	90°	90°
2	50	50	−90°	−90°
3	50	50	90°	90°
4	50	50	−90°	−90°
5	50	50	90°	90°
6	0	50	−90°	−90°

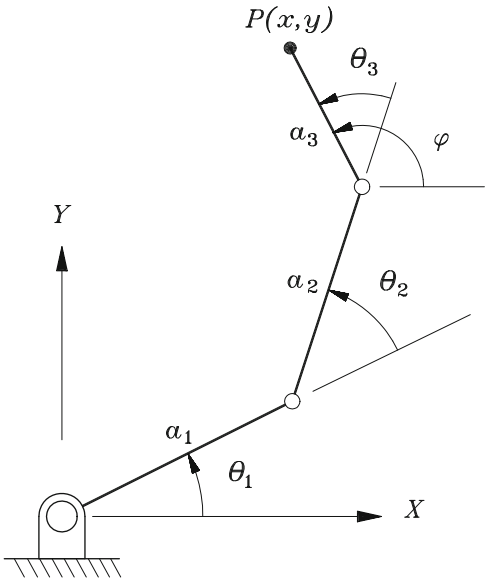


Fig. 5.21 A planar three-axis manipulator

5.2 The robotic manipulator of Fig. 4.23 is instrumented with sensors measuring the torque applied by the motors at the joints. Two readouts are taken of the sensors for the robot in the configuration indicated in the figure. In the first readout, the gripper is empty; in the second, it holds a tool. If the first readout is subtracted from the second, the vector difference $\Delta\boldsymbol{\tau}$ is obtained as

$$\Delta\boldsymbol{\tau} = \begin{bmatrix} 0 & 2 & 1 & 0 & 1 & 0 \end{bmatrix}^T \text{ Nm}$$

With the foregoing information, determine the weight w of the tool and the distance d of its mass center from C, the center of the spherical wrist.

5.3 A planar three-axis manipulator is shown in Fig. 5.21, with $a_1 = a_2 = a_3 = 1$ m. When a wrench acts onto the EE of this manipulator, the joint motors exert torques that keep the manipulator under static equilibrium. Readouts of these joint torques are recorded when the manipulator is in the posture $\theta_1 = \theta_2 = \theta_3 = 45°$, namely,

$$\tau_1 = -\sqrt{2}\,\text{Nm}, \qquad \tau_2 = -\sqrt{2}\,\text{Nm}, \qquad \tau_3 = 1 - \sqrt{2}\,\text{Nm}$$

Calculate the above-mentioned wrench.

5.4 For the two postures found in Exercise 4.8, the EE is to move with an angular velocity $\boldsymbol{\omega} = [\omega_1,\ \omega_2,\ \omega_3]^T\ \mathrm{s}^{-1}$. Show that if $\|\boldsymbol{\omega}\|$ remains constant, then so does $\|\dot{\boldsymbol{\theta}}\|$, for $\dot{\boldsymbol{\theta}}$ defined as the joint-rate vector of the wrist.

5.5 Point C of the manipulator of Fig. 4.17 is to move with a velocity $\mathbf{v}$ in the posture displayed in that figure. Show that as long as $\|\mathbf{v}\|$ remains constant, so does $\|\dot{\boldsymbol{\theta}}\|$, for $\dot{\boldsymbol{\theta}}$ defined as the joint-rate vector. Moreover, let us assume that in the same posture, point C is to attain a given acceleration $\mathbf{a}$. In general, however, $\|\ddot{\boldsymbol{\theta}}\|$, where $\ddot{\boldsymbol{\theta}}$ denotes the corresponding joint-acceleration vector, does not necessarily remain constant under a constant $\|\mathbf{a}\|$. Under which conditions does $\|\mathbf{a}\|$ remain constant for a constant $\|\ddot{\boldsymbol{\theta}}\|$?

5.6 A load $\mathbf{f}$ is applied to the manipulator of Fig. 4.17 in the posture displayed in that figure. Torque cells at the joints are calibrated to supply torque readouts resulting from this load only, and not from the dead load—its own weight—of the manipulator. Show that under a constant-magnitude load, the magnitude of the joint-torque vector remains constant as well.

5.7 Shown in Fig. 4.24 is the kinematic chain of an industrial robot, like the ABB-IRB 1000, which contains five revolutes and one prismatic pair.

(a) Determine the manipulator Jacobian in the X_1, Y_1, Z_1 coordinate frame fixed to the base.
(b) Determine the twist of the end-effector, defined in terms of the velocity of point P, for unit values of all joint-rates, and the posture displayed in the same figure.
(c) Determine the joint accelerations that will produce a vanishing acceleration of the point of intersection, C, of the three wrist axes and a vanishing angular acceleration of the gripper, for the unit joint rates given above.

5.8 The robot in Fig. 4.24 is now used for a deburring task. When the robot is in the configuration shown in that figure, a static force $\mathbf{f}$ and no moment acts on point P of the deburring tool. This force is sensed by torque sensors placed at the joints of the robot. Assume that the distance between the operating point P and the wrist center is 500 mm and that the readings of the arm joints are $\tau_1 = 0$, $\tau_2 = 100$ N m, and $\tau_3 = 50$ N m.

(a) Find the force $\mathbf{f}$ acting at P.
(b) Find the readings of the torque sensors placed at the wrist joints.

5.9 A decoupled manipulator is shown in Fig. 10.3 with the DH parameters of Table 10.1 at arbitrary posture.

(a) Find the Jacobian matrix of this manipulator at a posture with axis X_1 vertical and pointing downwards, while Z_2 and Y_1 make an angle of 180°. Moreover, in this particular posture, Z_3 and Z_4 are vertical and pointing upwards, while Z_7 makes an angle of 180° with Y_1.
(b) At the posture described in item (a), compute the joint-rates that will produce the twist

$$[\boldsymbol{\omega}]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \omega, \quad [\dot{\mathbf{p}}]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} v$$

(c) A wrench given by a moment $\mathbf{n}$ and a force $\mathbf{f}$ applied at point P acts on the EE of the same manipulator at the posture described in item (a) above. Calculate the joint torques or moments required to balance this wrench, which is given by

$$[\mathbf{n}]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} T, \quad [\mathbf{f}]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} F$$

5.10 A robot of the Puma type has the architecture displayed in Fig. 4.3, with the numerical values $a_2 = 0.432$ m, $a_3 = 0.020$ m, $b_3 = 0.149$ m, $b_4 = 0.432$ m. Find its maximum reach R as well as the link length a of the manipulator of Fig. 4.17 with the same reach R.

5.11 **Dialytic elimination**. The characteristic polynomial of decoupled manipulators for positioning tasks can be derived alternatively via *dialytic elimination*, as introduced in Sect. 5.4.1. It is recalled here that *dialytic elimination* consists in eliminating one unknown from a system of polynomial equations by expressing this system in *linear homogeneous form*, whereby each equation is a linear combination of various successive powers of the unknown to be eliminated, including the zeroth power. This elimination can be achieved as outlined below: In Eqs. (4.19a) and (4.20a), express $\cos\theta_1$ and $\sin\theta_1$ in terms of $\tan(\theta_1/2) \equiv t_1$, thereby obtaining

$$(-A + C\,c_3 + D\,s_3 + E)\,t_1^2 + 2\,B\,t_1 + (C\,c_3 + D\,s_3 + E + A) = 0$$
$$(H\,c_3 + I\,s_3 + J)\,t_1^2 + (2\,G - F)\,t_1 + (H\,c_3 + I\,s_3 + J + F) = 0$$

which can be further expressed as

$$m\,t_1^2 + 2\,B\,t_1 + n = 0$$
$$p\,t_1^2 + (2\,G - F)\,t_1 + q = 0$$

with obvious definitions for coefficients m, n, p, and q. Next, both sides of the two foregoing equations are multiplied by t_1, thereby producing

$$m\,t_1^3 + 2\,B\,t_1^2 + n\,t_1 = 0$$
$$p\,t_1^3 + (2\,G - F)\,t_1^2 + q\,t_1 = 0$$

Now, the last four equations can be regarded as a system of linear homogeneous equations, namely,

$$\mathbf{M}\mathbf{t}_1 = \mathbf{0}$$

where $\mathbf{0}$ is the four-dimensional zero vector, while $\mathbf{M}$ is a 4×4 matrix, and $\mathbf{t}_1$ is a four-dimensional vector. These arrays are defined as

$$\mathbf{M} \equiv \begin{bmatrix} 0 & m & 2B & n \\ 0 & p & 2G - F & q \\ m & 2B & n & 0 \\ p & 2G - F & q & 0 \end{bmatrix}, \quad \mathbf{t}_1 \equiv \begin{bmatrix} t_1^3 \\ t_1^2 \\ t_1 \\ 1 \end{bmatrix}$$

Apparently, $\mathbf{t}_1 \neq \mathbf{0}$, and hence, $\mathbf{M}$ must be singular. The characteristic polynomial sought can then be derived from the condition

$$\det(\mathbf{M}) = 0$$

Show that the last equation is quadratic in $\cos\theta_3$ and $\sin\theta_3$. Hence, the foregoing equation should lead to a quartic equation in $\tan(\theta_3/2)$. Derive the quartic equation involved. *Hint: Do not do this by hand, for it may be too time-consuming and could quickly lead to algebraic mistakes. Use software for symbolic computations instead.*

5.12 Compute the workspace volume of the manipulator of Fig. 4.3. Here, you can exploit the axial symmetry of the workspace by recalling the *Pappus–Guldinus Theorems*—see any book on multivariable calculus—that yield the volume as $2\pi q$, with q defined as the first moment of the cross-section, which is displayed in Fig. 5.4b, with respect to the axis of symmetry, Z_1. To this end, you will need the first moment of a semicircle with respect to its diameter. This information is tabulated in books on elementary mechanics or multivariable calculus, a.k.a. advanced calculus.

5.13 Compute the workspace volume of the manipulator of Fig. 4.17, whose workspace is sketched in Fig. 5.5. Here, you can also use the Pappus–Guldinus Theorem, as suggested in Exercise 5.12. However, the first moment of the cross-section has to be determined numerically, for the area properties of the curve that generates the three-dimensional workspace are not tabulated. Now, for two manipulators, the Puma-type and the one under discussion, with the same reach, determine which one has the larger workspace. *Note: This exercise is not more difficult than others, but it requires the use of suitable software for the calculation of area properties of planar regions bounded by arbitrary curves. Unless you have access to such software, do not attempt this exercise.*

5.14 Show that the maximum manipulability $\mu = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ of an orthogonal spherical wrist is attained when all three of its axes are mutually orthogonal. Find that maximum value.

5.15 Find an expression for the condition number of a three-revolute spherical wrist of twist angles α_4 and α_5, and show that this number depends only on α_4, α_5, and the intermediate joint angle, θ_5. Moreover, find values of these variables that minimize the condition number of the manipulator. *Hint: To find the required expression, the use of the condition number based on the*

Frobenius norm is strongly recommended. However, rendering the Jacobian matrix isotropic can be done by inspection.

5.16 **Manipulability of decoupled manipulators**. Let μ_a and μ_w represent the manipulability of the arm and the wrist of a decoupled manipulator, i.e.,

$$\mu_a \equiv \sqrt{\det(\mathbf{J}_{21}\mathbf{J}_{21}^T)}, \quad \mu_w \equiv \sqrt{\det(\mathbf{J}_{12}\mathbf{J}_{12}^T)}$$

with $\mathbf{J}_{12}$ and $\mathbf{J}_{21}$ defined in Sect. 5.2. Show that the manipulability μ of the overall manipulator is the product of the two manipulabilities given above, i.e.,

$$\mu = \mu_a \mu_w$$

5.17 Consider a planar two-revolute manipulator with link lengths a_1 and a_2. Find an expression of the form $\kappa(r, \theta_2)$ for the condition number of its Jacobian, with $r = a_2/a_1$, and establish values of r and θ_2 that minimize κ, which reaches a minimum value of unity.

5.18 Shown in Fig. 5.12 is an orthogonal three-revolute manipulator with an isotropic Jacobian. Find the volume of its workspace. Now consider a second manipulator with a similar orthogonal architecture, but with more common dimensions, i.e., with links of equal length λ. If the two manipulators have the same reach, that is, if

$$\lambda = \frac{1 + \sqrt{2}}{2} l$$

find the volume of the workspace of the second manipulator. Finally, determine the KCI—see Sect. 5.8 for a definition of this term—of the second manipulator. Draw some conclusions with regard to the performance of the two manipulators.