3

Differential Kinematics and Statics

In the previous chapter, direct and inverse kinematics equations establishing the relationship between the joint variables and the end-effector pose were derived. In this chapter, *differential kinematics* is presented which gives the relationship between the joint velocities and the corresponding end-effector linear and angular velocity. This mapping is described by a matrix, termed *geometric Jacobian*, which depends on the manipulator configuration. Alternatively, if the end-effector pose is expressed with reference to a minimal representation in the operational space, it is possible to compute the Jacobian matrix via differentiation of the direct kinematics function with respect to the joint variables. The resulting Jacobian, termed *analytical Jacobian*, in general differs from the geometric one. The Jacobian constitutes one of the most important tools for manipulator characterization; in fact, it is useful for finding *singularities*, analyzing *redundancy*, determining *inverse kinematics algorithms*, describing the mapping between forces applied to the end-effector and resulting torques at the joints (*statics*) and, as will be seen in the following chapters, deriving dynamic equations of motion and designing operational space control schemes. Finally, the *kineto-statics duality* concept is illustrated, which is at the basis of the definition of velocity and force *manipulability ellipsoids*.

3.1 Geometric Jacobian

Consider an n-DOF manipulator. The direct kinematics equation can be written in the form

$$\boldsymbol{T}_e(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R}_e(\boldsymbol{q}) & \boldsymbol{p}_e(\boldsymbol{q}) \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{q} = [\, q_1 \quad \dots \quad q_n \,]^T$ is the vector of joint variables. Both end-effector position and orientation vary as $\boldsymbol{q}$ varies.

The goal of the differential kinematics is to find the relationship between the joint velocities and the end-effector linear and angular velocities. In other words, it is desired to express the end-effector linear velocity $\dot{\boldsymbol{p}}_e$ and angular velocity $\boldsymbol{\omega}_e$ as a function of the joint velocities $\dot{\boldsymbol{q}}$. As will be seen afterwards, the sought relations are both linear in the joint velocities, i.e.,

$$\dot{\boldsymbol{p}}_e = \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.2}$$

$$\boldsymbol{\omega}_e = \boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.3}$$

In (3.2) $\boldsymbol{J}_P$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *linear* velocity $\dot{\boldsymbol{p}}_e$, while in (3.3) $\boldsymbol{J}_O$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *angular* velocity $\boldsymbol{\omega}_e$. In compact form, (3.2), (3.3) can be written as

$$\boldsymbol{v}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.4}$$

which represents the manipulator *differential kinematics equation*. The $(6 \times n)$ matrix $\boldsymbol{J}$ is the manipulator *geometric Jacobian*

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P \\ \boldsymbol{J}_O \end{bmatrix}, \tag{3.5}$$

which in general is a function of the joint variables.

In order to compute the geometric Jacobian, it is worth recalling a number of properties of rotation matrices and some important results of rigid body kinematics.

3.1.1 Derivative of a Rotation Matrix

The manipulator direct kinematics equation in (3.1) describes the end-effector pose, as a function of the joint variables, in terms of a position vector and a rotation matrix. Since the aim is to characterize the end-effector linear and angular velocities, it is worth considering first the *derivative of a rotation matrix* with respect to time.

Consider a time-varying rotation matrix $\boldsymbol{R} = \boldsymbol{R}(t)$. In view of the orthogonality of $\boldsymbol{R}$, one has the relation

$$\boldsymbol{R}(t)\boldsymbol{R}^T(t) = \boldsymbol{I}$$

which, differentiated with respect to time, gives the identity

$$\dot{\boldsymbol{R}}(t)\boldsymbol{R}^T(t) + \boldsymbol{R}(t)\dot{\boldsymbol{R}}^T(t) = \boldsymbol{O}.$$

Set

$$\boldsymbol{S}(t) = \dot{\boldsymbol{R}}(t)\boldsymbol{R}^T(t); \tag{3.6}$$

the (3×3) matrix $\boldsymbol{S}$ is *skew-symmetric* since

$$\boldsymbol{S}(t) + \boldsymbol{S}^T(t) = \boldsymbol{O}. \tag{3.7}$$

Postmultiplying both sides of (3.6) by $\boldsymbol{R}(t)$ gives

$$\dot{\boldsymbol{R}}(t) = \boldsymbol{S}(t)\boldsymbol{R}(t) \tag{3.8}$$

that allows the time derivative of $\boldsymbol{R}(t)$ to be expressed as a function of $\boldsymbol{R}(t)$ itself.

Equation (3.8) relates the rotation matrix $\boldsymbol{R}$ to its derivative by means of the skew-symmetric operator $\boldsymbol{S}$ and has a meaningful physical interpretation. Consider a constant vector $\boldsymbol{p}'$ and the vector $\boldsymbol{p}(t) = \boldsymbol{R}(t)\boldsymbol{p}'$. The time derivative of $\boldsymbol{p}(t)$ is

$$\dot{\boldsymbol{p}}(t) = \dot{\boldsymbol{R}}(t)\boldsymbol{p}',$$

which, in view of (3.8), can be written as

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{S}(t)\boldsymbol{R}(t)\boldsymbol{p}'.$$

If the vector $\boldsymbol{\omega}(t)$ denotes the *angular velocity* of frame $\boldsymbol{R}(t)$ with respect to the reference frame at time t, it is known from mechanics that

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{\omega}(t) \times \boldsymbol{R}(t)\boldsymbol{p}'.$$

Therefore, the matrix operator $\boldsymbol{S}(t)$ describes the vector product between the vector $\boldsymbol{\omega}$ and the vector $\boldsymbol{R}(t)\boldsymbol{p}'$. The matrix $\boldsymbol{S}(t)$ is so that its symmetric elements with respect to the main diagonal represent the components of the vector $\boldsymbol{\omega}(t) = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ in the form

$$\boldsymbol{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{3.9}$$

which justifies the expression $\boldsymbol{S}(t) = \boldsymbol{S}(\boldsymbol{\omega}(t))$. Hence, (3.8) can be rewritten as

$$\dot{\boldsymbol{R}} = \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}. \tag{3.10}$$

Furthermore, if $\boldsymbol{R}$ denotes a rotation matrix, it can be shown that the following relation holds:

$$\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}^T = \boldsymbol{S}(\boldsymbol{R}\boldsymbol{\omega}) \tag{3.11}$$

which will be useful later (see Problem 3.1).

Example 3.1

Consider the elementary rotation matrix about axis z given in (2.6). If α is a function of time, by computing the time derivative of $\boldsymbol{R}_z(\alpha(t))$, (3.6) becomes

$$\begin{aligned}\boldsymbol{S}(t) &= \begin{bmatrix} -\dot{\alpha}\sin\alpha & -\dot{\alpha}\cos\alpha & 0 \\ \dot{\alpha}\cos\alpha & -\dot{\alpha}\sin\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\dot{\alpha} & 0 \\ \dot{\alpha} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \boldsymbol{S}(\boldsymbol{\omega}(t)).\end{aligned}$$

According to (3.9), it is

$$\boldsymbol{\omega} = [\,0 \quad 0 \quad \dot{\alpha}\,]^T$$

that expresses the angular velocity of the frame about axis z.

With reference to Fig. 2.11, consider the coordinate transformation of a point P from Frame 1 to Frame 0; in view of (2.38), this is given by

$$\boldsymbol{p}^0 = \boldsymbol{o}_1^0 + \boldsymbol{R}_1^0 \boldsymbol{p}^1. \tag{3.12}$$

Differentiating (3.12) with respect to time gives

$$\dot{\boldsymbol{p}}^0 = \dot{\boldsymbol{o}}_1^0 + \boldsymbol{R}_1^0 \dot{\boldsymbol{p}}^1 + \dot{\boldsymbol{R}}_1^0 \boldsymbol{p}^1; \tag{3.13}$$

utilizing the expression of the derivative of a rotation matrix (3.8) and specifying the dependence on the angular velocity gives

$$\dot{\boldsymbol{p}}^0 = \dot{\boldsymbol{o}}_1^0 + \boldsymbol{R}_1^0 \dot{\boldsymbol{p}}^1 + \boldsymbol{S}(\boldsymbol{\omega}_1^0)\boldsymbol{R}_1^0 \boldsymbol{p}^1.$$

Further, denoting the vector $\boldsymbol{R}_1^0 \boldsymbol{p}^1$ by $\boldsymbol{r}_1^0$, it is

$$\dot{\boldsymbol{p}}^0 = \dot{\boldsymbol{o}}_1^0 + \boldsymbol{R}_1^0 \dot{\boldsymbol{p}}^1 + \boldsymbol{\omega}_1^0 \times \boldsymbol{r}_1^0 \tag{3.14}$$

which is the known form of the velocity composition rule.

Notice that, if $\boldsymbol{p}^1$ is *fixed* in Frame 1, then it is

$$\dot{\boldsymbol{p}}^0 = \dot{\boldsymbol{o}}_1^0 + \boldsymbol{\omega}_1^0 \times \boldsymbol{r}_1^0 \tag{3.15}$$

since $\dot{\boldsymbol{p}}^1 = \boldsymbol{0}$.

3.1.2 Link Velocities

Consider the generic Link i of a manipulator with an open kinematic chain. According to the Denavit–Hartenberg convention adopted in the previous chapter, Link i connects Joints i and $i+1$; Frame i is attached to Link i

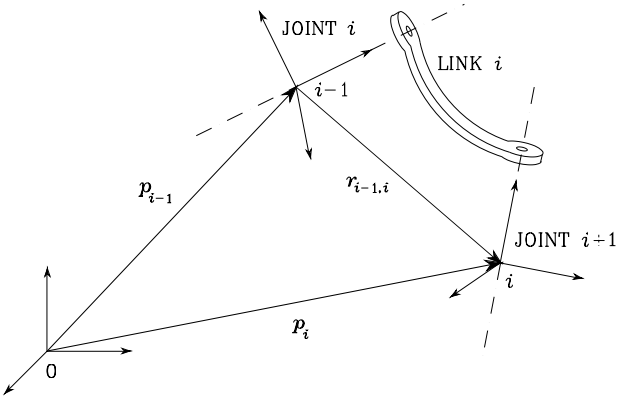

Fig. 3.1. Characterization of generic Link i of a manipulator

and has origin along Joint $i+1$ axis, while Frame $i-1$ has origin along Joint i axis (Fig. 3.1).

Let $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_i$ be the position vectors of the origins of Frames $i-1$ and i, respectively. Also, let $\boldsymbol{r}_{i-1,i}^{i-1}$ denote the position of the origin of Frame i with respect to Frame $i-1$ expressed in Frame $i-1$. According to the coordinate transformation (3.12), one can write[1]

$$\boldsymbol{p}_i = \boldsymbol{p}_{i-1} + \boldsymbol{R}_{i-1}\boldsymbol{r}_{i-1,i}^{i-1}.$$

Then, by virtue of (3.14), it is

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{R}_{i-1}\dot{\boldsymbol{r}}_{i-1,i}^{i-1} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{R}_{i-1}\boldsymbol{r}_{i-1,i}^{i-1} = \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{v}_{i-1,i} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{r}_{i-1,i} \tag{3.16}$$

which gives the expression of the linear velocity of Link i as a function of the translational and rotational velocities of Link $i-1$. Note that $\boldsymbol{v}_{i-1,i}$ denotes the velocity of the origin of Frame i with respect to the origin of Frame $i-1$.

Concerning link angular velocity, it is worth starting from the rotation composition

$$\boldsymbol{R}_i = \boldsymbol{R}_{i-1}\boldsymbol{R}_i^{i-1};$$

from (3.8), its time derivative can be written as

$$\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{R}_i = \boldsymbol{S}(\boldsymbol{\omega}_{i-1})\boldsymbol{R}_i + \boldsymbol{R}_{i-1}\boldsymbol{S}(\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_i^{i-1} \tag{3.17}$$

where $\boldsymbol{\omega}_{i-1,i}^{i-1}$ denotes the angular velocity of Frame i with respect to Frame $i-1$ expressed in Frame $i-1$. From (2.4), the second term on the right-hand side of (3.17) can be rewritten as

$$\boldsymbol{R}_{i-1}\boldsymbol{S}(\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_i^{i-1} = \boldsymbol{R}_{i-1}\boldsymbol{S}(\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_{i-1}^T\boldsymbol{R}_{i-1}\boldsymbol{R}_i^{i-1};$$

[1] Hereafter, the indication of superscript '0' is omitted for quantities referred to Frame 0. Also, without loss of generality, Frame 0 and Frame n are taken as the base frame and the end-effector frame, respectively.

in view of property (3.11), it is

$$\boldsymbol{R}_{i-1}\boldsymbol{S}(\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_i^{i-1} = \boldsymbol{S}(\boldsymbol{R}_{i-1}\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_i.$$

Then, (3.17) becomes

$$\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{R}_i = \boldsymbol{S}(\boldsymbol{\omega}_{i-1})\boldsymbol{R}_i + \boldsymbol{S}(\boldsymbol{R}_{i-1}\boldsymbol{\omega}_{i-1,i}^{i-1})\boldsymbol{R}_i$$

leading to the result

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \boldsymbol{R}_{i-1}\boldsymbol{\omega}_{i-1,i}^{i-1} = \boldsymbol{\omega}_{i-1} + \boldsymbol{\omega}_{i-1,i}, \tag{3.18}$$

which gives the expression of the angular velocity of Link i as a function of the angular velocities of Link $i-1$ and of Link i with respect to Link $i-1$.

The relations (3.16), (3.18) attain different expressions depending on the type of Joint i (*prismatic* or *revolute*).

Prismatic joint

Since orientation of Frame i with respect to Frame $i-1$ does not vary by moving Joint i, it is

$$\boldsymbol{\omega}_{i-1,i} = \boldsymbol{0}. \tag{3.19}$$

Further, the linear velocity is

$$\boldsymbol{v}_{i-1,i} = \dot{d}_i\boldsymbol{z}_{i-1} \tag{3.20}$$

where $\boldsymbol{z}_{i-1}$ is the unit vector of Joint i axis. Hence, the expressions of angular velocity (3.18) and linear velocity (3.16) respectively become

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} \tag{3.21}$$

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \dot{d}_i\boldsymbol{z}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}, \tag{3.22}$$

where the relation $\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1}$ has been exploited to derive (3.22).

Revolute joint

For the angular velocity it is obviously

$$\boldsymbol{\omega}_{i-1,i} = \dot{\vartheta}_i\boldsymbol{z}_{i-1}, \tag{3.23}$$

while for the linear velocity it is

$$\boldsymbol{v}_{i-1,i} = \boldsymbol{\omega}_{i-1,i} \times \boldsymbol{r}_{i-1,i} \tag{3.24}$$

due to the rotation of Frame i with respect to Frame $i-1$ induced by the motion of Joint i. Hence, the expressions of angular velocity (3.18) and linear velocity (3.16) respectively become

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\vartheta}_i\boldsymbol{z}_{i-1} \tag{3.25}$$

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}, \tag{3.26}$$

where (3.18) has been exploited to derive (3.26).

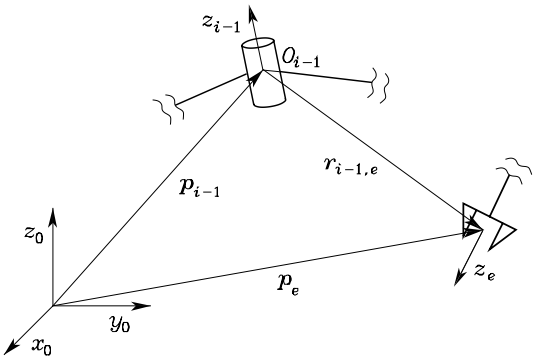

Fig. 3.2. Representation of vectors needed for the computation of the velocity contribution of a revolute joint to the end-effector linear velocity

3.1.3 Jacobian Computation

In order to compute the Jacobian, it is convenient to proceed separately for the linear velocity and the angular velocity.

For the contribution to the *linear velocity*, the time derivative of $\boldsymbol{p}_e(\boldsymbol{q})$ can be written as

$$\dot{\boldsymbol{p}}_e = \sum_{i=1}^{n} \frac{\partial \boldsymbol{p}_e}{\partial q_i} \dot{q}_i = \sum_{i=1}^{n} \boldsymbol{\jmath}_{Pi} \dot{q}_i. \tag{3.27}$$

This expression shows how $\dot{\boldsymbol{p}}_e$ can be obtained as the sum of the terms $\dot{q}_i \boldsymbol{\jmath}_{Pi}$. Each term represents the contribution of the velocity of single Joint i to the end-effector linear velocity when all the other joints are still.

Therefore, by distinguishing the case of a *prismatic* joint ($q_i = d_i$) from the case of a *revolute* joint ($q_i = \vartheta_i$), it is:

- If Joint i is *prismatic*, from (3.20) it is

$$\dot{q}_i \boldsymbol{\jmath}_{Pi} = \dot{d}_i \boldsymbol{z}_{i-1}$$

and then

$$\boldsymbol{\jmath}_{Pi} = \boldsymbol{z}_{i-1}.$$

- If Joint i is *revolute*, observing that the contribution to the linear velocity is to be computed with reference to the origin of the end-effector frame (Fig. 3.2), it is

$$\dot{q}_i \boldsymbol{\jmath}_{Pi} = \boldsymbol{\omega}_{i-1,i} \times \boldsymbol{r}_{i-1,e} = \dot{\vartheta}_i \boldsymbol{z}_{i-1} \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1})$$

and then

$$\boldsymbol{\jmath}_{Pi} = \boldsymbol{z}_{i-1} \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1}).$$

For the contribution to the *angular velocity*, in view of (3.18), it is

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_n = \sum_{i=1}^{n} \boldsymbol{\omega}_{i-1,i} = \sum_{i=1}^{n} \boldsymbol{\jmath}_{Oi} \dot{q}_i, \tag{3.28}$$

where (3.19) and (3.23) have been utilized to characterize the terms $\dot{q}_i \boldsymbol{\jmath}_{Oi}$, and thus in detail:

- If Joint i is *prismatic*, from (3.19) it is

$$\dot{q}_i \boldsymbol{\jmath}_{Oi} = \mathbf{0}$$

and then

$$\boldsymbol{\jmath}_{Oi} = \mathbf{0}.$$

- If Joint i is *revolute*, from (3.23) it is

$$\dot{q}_i \boldsymbol{\jmath}_{Oi} = \dot{\vartheta}_i \boldsymbol{z}_{i-1}$$

and then

$$\boldsymbol{\jmath}_{Oi} = \boldsymbol{z}_{i-1}.$$

In summary, the Jacobian in (3.5) can be partitioned into the (3×1) column vectors $\boldsymbol{\jmath}_{Pi}$ and $\boldsymbol{\jmath}_{Oi}$ as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{\jmath}_{P1} & \cdots & \boldsymbol{\jmath}_{Pn} \\ \boldsymbol{\jmath}_{O1} & & \boldsymbol{\jmath}_{On} \end{bmatrix}, \tag{3.29}$$

where

$$\begin{bmatrix} \boldsymbol{\jmath}_{Pi} \\ \boldsymbol{\jmath}_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} \boldsymbol{z}_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} \boldsymbol{z}_{i-1} \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1}) \\ \boldsymbol{z}_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{3.30}$$

The expressions in (3.30) allow Jacobian computation in a simple, systematic way on the basis of direct kinematics relations. In fact, the vectors $\boldsymbol{z}_{i-1}$, $\boldsymbol{p}_e$ and $\boldsymbol{p}_{i-1}$ are all functions of the joint variables. In particular:

- $\boldsymbol{z}_{i-1}$ is given by the third column of the rotation matrix $\boldsymbol{R}_{i-1}^0$, i.e.,

$$\boldsymbol{z}_{i-1} = \boldsymbol{R}_1^0(q_1) \dots \boldsymbol{R}_{i-1}^{i-2}(q_{i-1}) \boldsymbol{z}_0 \tag{3.31}$$

  where $\boldsymbol{z}_0 = [\,0 \quad 0 \quad 1\,]^T$ allows the selection of the third column.
- $\boldsymbol{p}_e$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_e^0$, i.e., by expressing $\widetilde{\boldsymbol{p}}_e$ in the (4×1) homogeneous form

$$\widetilde{\boldsymbol{p}}_e = \boldsymbol{A}_1^0(q_1) \dots \boldsymbol{A}_n^{n-1}(q_n) \widetilde{\boldsymbol{p}}_0 \tag{3.32}$$

  where $\widetilde{\boldsymbol{p}}_0 = [\,0 \quad 0 \quad 0 \quad 1\,]^T$ allows the selection of the fourth column.

- $\boldsymbol{p}_{i-1}$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_{i-1}^{0}$, i.e., it can be extracted from

$$\widetilde{\boldsymbol{p}}_{i-1} = \boldsymbol{A}_1^0(q_1)\dots\boldsymbol{A}_{i-1}^{i-2}(q_{i-1})\widetilde{\boldsymbol{p}}_0. \tag{3.33}$$

The above equations can be conveniently used to compute the translational and rotational velocities of any point along the manipulator structure, as long as the direct kinematics functions relative to that point are known.

Finally, notice that the Jacobian matrix depends on the frame in which the end-effector velocity is expressed. The above equations allow computation of the geometric Jacobian with respect to the base frame. If it is desired to represent the Jacobian in a different Frame u, it is sufficient to know the relative rotation matrix $\boldsymbol{R}^u$. The relationship between velocities in the two frames is

$$\begin{bmatrix} \dot{\boldsymbol{p}}_e^u \\ \boldsymbol{\omega}_e^u \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^u & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{R}^u \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix},$$

which, substituted in (3.4), gives

$$\begin{bmatrix} \dot{\boldsymbol{p}}_e^u \\ \boldsymbol{\omega}_e^u \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^u & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{R}^u \end{bmatrix} \boldsymbol{J}\dot{\boldsymbol{q}}$$

and then

$$\boldsymbol{J}^u = \begin{bmatrix} \boldsymbol{R}^u & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{R}^u \end{bmatrix} \boldsymbol{J}, \tag{3.34}$$

where $\boldsymbol{J}^u$ denotes the geometric Jacobian in Frame u.

3.2 Jacobian of Typical Manipulator Structures

In the following, the Jacobian is computed for some of the typical manipulator structures presented in the previous chapter.

3.2.1 Three-link Planar Arm

In this case, from (3.30) the Jacobian is

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{p}_3 - \boldsymbol{p}_0) & \boldsymbol{z}_1 \times (\boldsymbol{p}_3 - \boldsymbol{p}_1) & \boldsymbol{z}_2 \times (\boldsymbol{p}_3 - \boldsymbol{p}_2) \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}.$$

Computation of the position vectors of the various links gives

$$\boldsymbol{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_2 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}$$

$$\boldsymbol{p}_3 = \begin{bmatrix} a_1c_1 + a_2c_{12} + a_3c_{123} \\ a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 \end{bmatrix}$$

while computation of the unit vectors of revolute joint axes gives

$$\boldsymbol{z}_0 = \boldsymbol{z}_1 = \boldsymbol{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

since they are all parallel to axis z_0. From (3.29) it is

$$\boldsymbol{J} = \begin{bmatrix} -a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} & -a_3s_{123} \\ a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} & a_3c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{3.35}$$

In the Jacobian (3.35), only the three non-null rows are relevant (the rank of the matrix is at most 3); these refer to the two components of linear velocity along axes x_0, y_0 and the component of angular velocity about axis z_0. This result can be derived by observing that three DOFs allow specification of at most three end-effector variables; v_z, ω_x, ω_y are always null for this kinematic structure. If orientation is of no concern, the (2×3) Jacobian for the positional part can be derived by considering just the first two rows, i.e.,

$$\boldsymbol{J}_P = \begin{bmatrix} -a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} & -a_3s_{123} \\ a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} & a_3c_{123} \end{bmatrix}. \tag{3.36}$$

3.2.2 Anthropomorphic Arm

In this case, from (3.30) the Jacobian is

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{p}_3 - \boldsymbol{p}_0) & \boldsymbol{z}_1 \times (\boldsymbol{p}_3 - \boldsymbol{p}_1) & \boldsymbol{z}_2 \times (\boldsymbol{p}_3 - \boldsymbol{p}_2) \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}.$$

Computation of the position vectors of the various links gives

$$\boldsymbol{p}_0 = \boldsymbol{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_2 = \begin{bmatrix} a_2c_1c_2 \\ a_2s_1c_2 \\ a_2s_2 \end{bmatrix}$$

$$\boldsymbol{p}_3 = \begin{bmatrix} c_1(a_2c_2 + a_3c_{23}) \\ s_1(a_2c_2 + a_3c_{23}) \\ a_2s_2 + a_3s_{23} \end{bmatrix}$$

while computation of the unit vectors of revolute joint axes gives

$$\boldsymbol{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \boldsymbol{z}_1 = \boldsymbol{z}_2 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}.$$

From (3.29) it is

$$\boldsymbol{J} = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.37}$$

Only three of the six rows of the Jacobian (3.37) are linearly independent. Having 3 DOFs only, it is worth considering the upper (3×3) block of the Jacobian

$$\boldsymbol{J}_P = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \end{bmatrix} \tag{3.38}$$

that describes the relationship between the joint velocities and the end-effector linear velocity. This structure does not allow an arbitrary angular velocity $\boldsymbol{\omega}$ to be obtained; in fact, the two components ω_x and ω_y are not independent $(s_1\omega_y = -c_1\omega_x)$.

3.2.3 Stanford Manipulator

In this case, from (3.30) it is

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{p}_6 - \boldsymbol{p}_0) & \boldsymbol{z}_1 \times (\boldsymbol{p}_6 - \boldsymbol{p}_1) & \boldsymbol{z}_2 & & \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{0} & & \\ & & \boldsymbol{z}_3 \times (\boldsymbol{p}_6 - \boldsymbol{p}_3) & \boldsymbol{z}_4 \times (\boldsymbol{p}_6 - \boldsymbol{p}_4) & \boldsymbol{z}_5 \times (\boldsymbol{p}_6 - \boldsymbol{p}_5) \\ & & \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 \end{bmatrix}.$$

Computation of the position vectors of the various links gives

$$\boldsymbol{p}_0 = \boldsymbol{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{p}_3 = \boldsymbol{p}_4 = \boldsymbol{p}_5 = \begin{bmatrix} c_1s_2d_3 - s_1d_2 \\ s_1s_2d_3 + c_1d_2 \\ c_2d_3 \end{bmatrix}$$

$$\boldsymbol{p}_6 = \begin{bmatrix} c_1s_2d_3 - s_1d_2 + \big(c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5\big)d_6 \\ s_1s_2d_3 + c_1d_2 + \big(s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5\big)d_6 \\ c_2d_3 + (-s_2c_4s_5 + c_2c_5)d_6 \end{bmatrix},$$

while computation of the unit vectors of joint axes gives

$$\boldsymbol{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \boldsymbol{z}_1 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \quad \boldsymbol{z}_2 = \boldsymbol{z}_3 = \begin{bmatrix} c_1 s_2 \\ s_1 s_2 \\ c_2 \end{bmatrix}$$

$$\boldsymbol{z}_4 = \begin{bmatrix} -c_1 c_2 s_4 - s_1 c_4 \\ -s_1 c_2 s_4 + c_1 c_4 \\ s_2 s_4 \end{bmatrix} \quad \boldsymbol{z}_5 = \begin{bmatrix} c_1(c_2 c_4 s_5 + s_2 c_5) - s_1 s_4 s_5 \\ s_1(c_2 c_4 s_5 + s_2 c_5) + c_1 s_4 s_5 \\ -s_2 c_4 s_5 + c_2 c_5 \end{bmatrix}.$$

The sought Jacobian can be obtained by developing the computations as in (3.29), leading to expressing end-effector linear and angular velocity as a function of joint velocities.

3.3 Kinematic Singularities

The Jacobian in the differential kinematics equation of a manipulator defines a linear mapping

$$\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.39}$$

between the vector $\dot{\boldsymbol{q}}$ of joint velocities and the vector $\boldsymbol{v}_e = [\,\dot{\boldsymbol{p}}_e^T \quad \boldsymbol{\omega}_e^T\,]^T$ of end-effector velocity. The Jacobian is, in general, a function of the configuration $\boldsymbol{q}$; those configurations at which $\boldsymbol{J}$ is rank-deficient are termed *kinematic singularities*. To find the singularities of a manipulator is of great interest for the following reasons:

a) Singularities represent configurations at which mobility of the structure is reduced, i.e., it is not possible to impose an arbitrary motion to the end-effector.
b) When the structure is at a singularity, infinite solutions to the inverse kinematics problem may exist.
c) In the neighbourhood of a singularity, small velocities in the operational space may cause large velocities in the joint space.

Singularities can be classified into:

- *Boundary* singularities that occur when the manipulator is either outstretched or retracted. It may be understood that these singularities do not represent a true drawback, since they can be avoided on condition that the manipulator is not driven to the boundaries of its reachable workspace.
- *Internal* singularities that occur inside the reachable workspace and are generally caused by the alignment of two or more axes of motion, or else by the attainment of particular end-effector configurations. Unlike the above, these singularities constitute a serious problem, as they can be encountered anywhere in the reachable workspace for a planned path in the operational space.

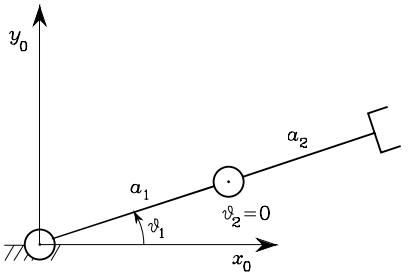

Fig. 3.3. Two-link planar arm at a boundary singularity

Example 3.2

To illustrate the behaviour of a manipulator at a singularity, consider a two-link planar arm. In this case, it is worth considering only the components $\dot{p}_x$ and $\dot{p}_y$ of the linear velocity in the plane. Thus, the Jacobian is the (2×2) matrix

$$\boldsymbol{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}. \tag{3.40}$$

To analyze matrix rank, consider its determinant given by

$$\det(\boldsymbol{J}) = a_1 a_2 s_2. \tag{3.41}$$

For $a_1, a_2 \neq 0$, it is easy to find that the determinant in (3.41) vanishes whenever

$$\vartheta_2 = 0 \qquad \vartheta_2 = \pi,$$

ϑ_1 being irrelevant for the determination of singular configurations. These occur when the arm tip is located either on the outer ($\vartheta_2 = 0$) or on the inner ($\vartheta_2 = \pi$) boundary of the reachable workspace. Figure 3.3 illustrates the arm posture for $\vartheta_2 = 0$.

By analyzing the differential motion of the structure in such configuration, it can be observed that the two column vectors $[\,-(a_1 + a_2)s_1 \quad (a_1 + a_2)c_1\,]^T$ and $[\,-a_2 s_1 \quad a_2 c_1\,]^T$ of the Jacobian become parallel, and thus the Jacobian rank becomes one; this means that the tip velocity components are not independent (see point **a)** above).

3.3.1 Singularity Decoupling

Computation of internal singularities via the Jacobian determinant may be tedious and of no easy solution for complex structures. For manipulators having a spherical wrist, by analogy with what has already been seen for inverse kinematics, it is possible to split the problem of singularity computation into two separate problems:

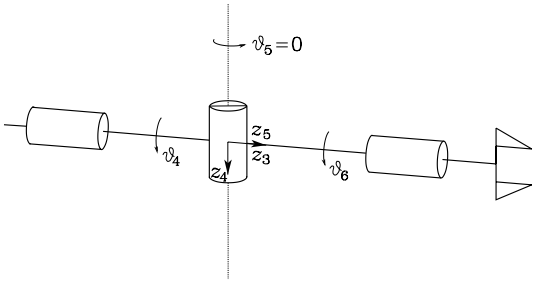

Fig. 3.4. Spherical wrist at a singularity

- computation of *arm singularities* resulting from the motion of the first 3 or more links,
- computation of *wrist singularities* resulting from the motion of the wrist joints.

For the sake of simplicity, consider the case $n = 6$; the Jacobian can be partitioned into (3×3) blocks as follows:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix} \tag{3.42}$$

where, since the outer 3 joints are all revolute, the expressions of the two right blocks are respectively

$$\boldsymbol{J}_{12} = \begin{bmatrix} \boldsymbol{z}_3 \times (\boldsymbol{p}_e - \boldsymbol{p}_3) & \boldsymbol{z}_4 \times (\boldsymbol{p}_e - \boldsymbol{p}_4) & \boldsymbol{z}_5 \times (\boldsymbol{p}_e - \boldsymbol{p}_5) \end{bmatrix}$$

$$\boldsymbol{J}_{22} = \begin{bmatrix} \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 \end{bmatrix}. \tag{3.43}$$

As singularities are typical of the mechanical structure and do not depend on the frames chosen to describe kinematics, it is convenient to choose the origin of the end-effector frame at the intersection of the wrist axes (see Fig. 2.32). The choice $\boldsymbol{p} = \boldsymbol{p}_W$ leads to

$$\boldsymbol{J}_{12} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

since all vectors $\boldsymbol{p}_W - \boldsymbol{p}_i$ are parallel to the unit vectors $\boldsymbol{z}_i$, for $i = 3, 4, 5$, no matter how Frames $3, 4, 5$ are chosen according to DH convention. In view of this choice, the overall Jacobian becomes a block lower-triangular matrix. In this case, computation of the determinant is greatly simplified, as this is given by the product of the determinants of the two blocks on the diagonal, i.e.,

$$\det(\boldsymbol{J}) = \det(\boldsymbol{J}_{11})\det(\boldsymbol{J}_{22}). \tag{3.44}$$

In turn, a true *singularity decoupling* has been achieved; the condition

$$\det(\boldsymbol{J}_{11}) = 0$$

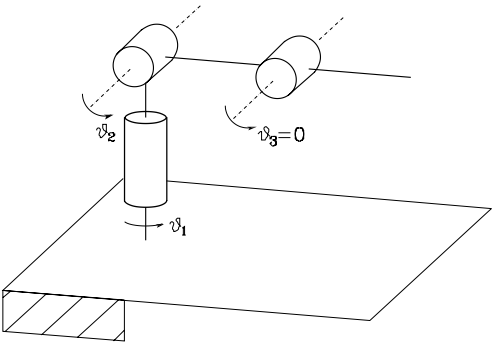

Fig. 3.5. Anthropomorphic arm at an elbow singularity

leads to determining the *arm singularities*, while the condition

$$\det(\boldsymbol{J}_{22}) = 0$$

leads to determining the *wrist singularities*.

Notice, however, that this form of Jacobian does not provide the relationship between the joint velocities and the end-effector velocity, but it leads to simplifying singularity computation. Below the two types of singularities are analyzed in detail.

3.3.2 Wrist Singularities

On the basis of the above singularity decoupling, wrist singularities can be determined by inspecting the block $\boldsymbol{J}_{22}$ in (3.43). It can be recognized that the wrist is at a singular configuration whenever the unit vectors $\boldsymbol{z}_3$, $\boldsymbol{z}_4$, $\boldsymbol{z}_5$ are linearly dependent. The wrist kinematic structure reveals that a singularity occurs when $\boldsymbol{z}_3$ and $\boldsymbol{z}_5$ are aligned, i.e., whenever

$$\vartheta_5 = 0 \qquad \vartheta_5 = \pi.$$

Taking into consideration only the first configuration (Fig. 3.4), the loss of mobility is caused by the fact that rotations of equal magnitude about opposite directions on ϑ_4 and ϑ_6 do not produce any end-effector rotation. Further, the wrist is not allowed to rotate about the axis orthogonal to $\boldsymbol{z}_4$ and $\boldsymbol{z}_3$, (see point **a)** above). This singularity is naturally described in the joint space and can be encountered anywhere inside the manipulator reachable workspace; as a consequence, special care is to be taken in programming an end-effector motion.

3.3.3 Arm Singularities

Arm singularities are characteristic of a specific manipulator structure; to illustrate their determination, consider the anthropomorphic arm (Fig. 2.23),

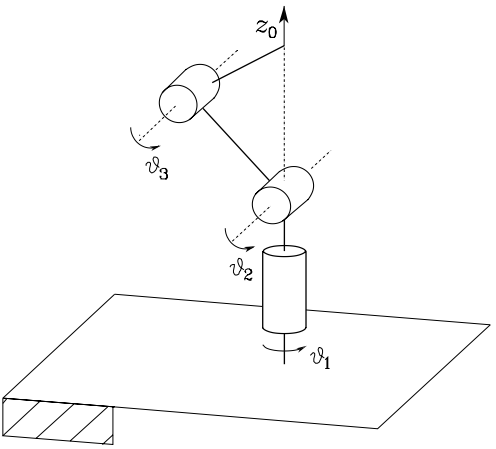

Fig. 3.6. Anthropomorphic arm at a shoulder singularity

whose Jacobian for the linear velocity part is given by (3.38). Its determinant is

$$\det(\boldsymbol{J}_P) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23}).$$

Like in the case of the planar arm of Example 3.2, the determinant does not depend on the first joint variable.

For $a_2, a_3 \neq 0$, the determinant vanishes if $s_3 = 0$ and/or $(a_2 c_2 + a_3 c_{23}) = 0$. The first situation occurs whenever

$$\vartheta_3 = 0 \qquad \vartheta_3 = \pi$$

meaning that the elbow is outstretched (Fig. 3.5) or retracted, and is termed *elbow singularity*. Notice that this type of singularity is conceptually equivalent to the singularity found for the two-link planar arm.

By recalling the direct kinematics equation in (2.66), it can be observed that the second situation occurs when the wrist point lies on axis z_0 (Fig. 3.6); it is thus characterized by

$$p_x = p_y = 0$$

and is termed *shoulder singularity*.

Notice that the whole axis z_0 describes a continuum of singular configurations; a rotation of ϑ_1 does not cause any translation of the wrist position (the first column of $\boldsymbol{J}_P$ is always null at a shoulder singularity), and then the inverse kinematics equation admits infinite solutions; moreover, motions starting from the singular configuration that take the wrist along the z_1 direction are not allowed (see point **b)** above).

If a spherical wrist is connected to an anthropomorphic arm (Fig. 2.26), the arm direct kinematics is different. In this case the Jacobian to consider represents the block $\boldsymbol{J}_{11}$ of the Jacobian in (3.42) with $\boldsymbol{p} = \boldsymbol{p}_W$. Analyzing its

determinant leads to finding the same singular configurations, which are relative to different values of the third joint variables, though — compare (2.66) and (2.70).

Finally, it is important to remark that, unlike the wrist singularities, the arm singularities are well identified in the operational space, and thus they can be suitably avoided in the end-effector trajectory planning stage.

3.4 Analysis of Redundancy

The concept of *kinematic redundancy* has been introduced in Sect. 2.10.2; redundancy is related to the number n of DOFs of the structure, the number m of operational space variables, and the number r of operational space variables necessary to specify a given task.

In order to perform a systematic analysis of redundancy, it is worth considering differential kinematics in lieu of direct kinematics (2.82). To this end, (3.39) is to be interpreted as the differential kinematics mapping relating the n components of the joint velocity vector to the $r \leq m$ components of the velocity vector $\boldsymbol{v}_e$ of concern for the specific task. To clarify this point, consider the case of a 3-link planar arm; that is not intrinsically redundant ($n = m = 3$) and its Jacobian (3.35) has 3 null rows accordingly. If the task does not specify ω_z ($r = 2$), the arm becomes functionally redundant and the Jacobian to consider for redundancy analysis is the one in (3.36).

A different case is that of the anthropomorphic arm for which only position variables are of concern ($n = m = 3$). The relevant Jacobian is the one in (3.38). The arm is neither intrinsically redundant nor can become functionally redundant if it is assigned a planar task; in that case, indeed, the task would set constraints on the 3 components of end-effector linear velocity.

Therefore, the differential kinematics equation to consider can be formally written as in (3.39), i.e.,

$$\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3.45}$$

where now $\boldsymbol{v}_e$ is meant to be the $(r \times 1)$ vector of end-effector velocity of concern for the specific task and $\boldsymbol{J}$ is the corresponding $(r \times n)$ Jacobian matrix that can be extracted from the geometric Jacobian; $\dot{\boldsymbol{q}}$ is the $(n \times 1)$ vector of joint velocities. If $r < n$, the manipulator is kinematically redundant and there exist $(n - r)$ *redundant DOFs*.

The Jacobian describes the linear mapping from the joint velocity space to the end-effector velocity space. In general, it is a function of the configuration. In the context of differential kinematics, however, the Jacobian has to be regarded as a constant matrix, since the instantaneous velocity mapping is of interest for a given posture. The mapping is schematically illustrated in Fig. 3.7 with a typical notation from set theory.

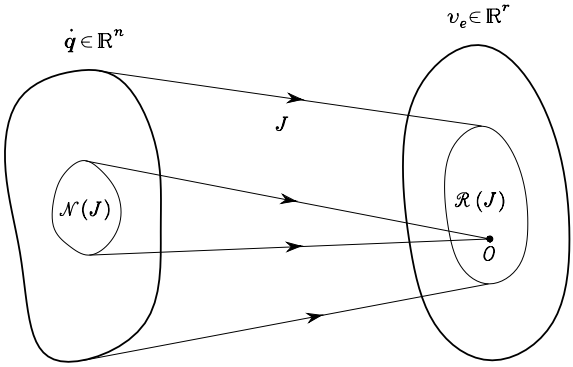

Fig. 3.7. Mapping between the joint velocity space and the end-effector velocity space

The differential kinematics equation in (3.45) can be characterized in terms of the *range* and *null* spaces of the mapping;[2] specifically, one has that:

- The *range* space of $\boldsymbol{J}$ is the subspace $\mathcal{R}(\boldsymbol{J})$ in $\mathbb{R}^r$ of the end-effector velocities that can be generated by the joint velocities, in the given manipulator posture.
- The *null* space of $\boldsymbol{J}$ is the subspace $\mathcal{N}(\boldsymbol{J})$ in $\mathbb{R}^n$ of joint velocities that do not produce any end-effector velocity, in the given manipulator posture.

If the Jacobian has *full rank*, one has

$$\dim\big(\mathcal{R}(\boldsymbol{J})\big) = r \qquad \dim\big(\mathcal{N}(\boldsymbol{J})\big) = n - r$$

and the range of $\boldsymbol{J}$ spans the entire space $\mathbb{R}^r$. Instead, if the Jacobian degenerates at a *singularity*, the dimension of the range space decreases while the dimension of the null space increases, since the following relation holds:

$$\dim\big(\mathcal{R}(\boldsymbol{J})\big) + \dim\big(\mathcal{N}(\boldsymbol{J})\big) = n$$

independently of the rank of the matrix $\boldsymbol{J}$.

The existence of a subspace $\mathcal{N}(\boldsymbol{J}) \neq \emptyset$ for a redundant manipulator allows determination of systematic techniques for handling redundant DOFs. To this end, if $\dot{\boldsymbol{q}}^*$ denotes a solution to (3.45) and $\boldsymbol{P}$ is an $(n \times n)$ matrix so that

$$\mathcal{R}(\boldsymbol{P}) \equiv \mathcal{N}(\boldsymbol{J}),$$

the joint velocity vector

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^* + \boldsymbol{P}\dot{\boldsymbol{q}}_0, \tag{3.46}$$

with arbitrary $\dot{\boldsymbol{q}}_0$, is also a solution to (3.45). In fact, premultiplying both sides of (3.46) by $\boldsymbol{J}$ yields

$$\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}}^* + \boldsymbol{J}\boldsymbol{P}\dot{\boldsymbol{q}}_0 = \boldsymbol{J}\dot{\boldsymbol{q}}^* = \boldsymbol{v}_e$$

[2] See Sect. A.4 for the linear mappings.

since $\boldsymbol{J}\boldsymbol{P}\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$ for any $\dot{\boldsymbol{q}}_0$. This result is of fundamental importance for redundancy resolution; a solution of the kind (3.46) points out the possibility of choosing the vector of arbitrary joint velocities $\dot{\boldsymbol{q}}_0$ so as to exploit advantageously the redundant DOFs. In fact, the effect of $\dot{\boldsymbol{q}}_0$ is to generate *internal motions* of the structure that do not change the end-effector position and orientation but may allow, for instance, manipulator reconfiguration into more dexterous postures for execution of a given task.

3.5 Inverse Differential Kinematics

In Sect. 2.12 it was shown how the inverse kinematics problem admits closed-form solutions only for manipulators having a simple kinematic structure. Problems arise whenever the end-effector attains a particular position and/or orientation in the operational space, or the structure is complex and it is not possible to relate the end-effector pose to different sets of joint variables, or else the manipulator is redundant. These limitations are caused by the highly nonlinear relationship between joint space variables and operational space variables.

On the other hand, the differential kinematics equation represents a linear mapping between the joint velocity space and the operational velocity space, although it varies with the current configuration. This fact suggests the possibility to utilize the differential kinematics equation to tackle the inverse kinematics problem.

Suppose that a motion trajectory is assigned to the end-effector in terms of $\boldsymbol{v}_e$ and the initial conditions on position and orientation. The aim is to determine a feasible joint trajectory $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$ that reproduces the given trajectory.

By considering (3.45) with $n = r$, the joint velocities can be obtained via simple inversion of the Jacobian matrix

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\boldsymbol{v}_e. \tag{3.47}$$

If the initial manipulator posture $\boldsymbol{q}(0)$ is known, joint positions can be computed by integrating velocities over time, i.e.,

$$\boldsymbol{q}(t) = \int_0^t \dot{\boldsymbol{q}}(\varsigma)d\varsigma + \boldsymbol{q}(0).$$

The integration can be performed in discrete time by resorting to numerical techniques. The simplest technique is based on the Euler integration method; given an integration interval Δt, if the joint positions and velocities at time t_k are known, the joint positions at time $t_{k+1} = t_k + \Delta t$ can be computed as

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{q}(t_k) + \dot{\boldsymbol{q}}(t_k)\Delta t. \tag{3.48}$$

This technique for inverting kinematics is independent of the solvability of the kinematic structure. Nonetheless, it is necessary that the *Jacobian* be *square* and of *full rank*; this demands further insight into the cases of *redundant* manipulators and kinematic *singularity* occurrence.

3.5.1 Redundant Manipulators

When the manipulator is *redundant* ($r < n$), the Jacobian matrix has more columns than rows and infinite solutions exist to (3.45). A viable solution method is to formulate the problem as a constrained linear optimization problem.

In detail, once the end-effector velocity $\boldsymbol{v}_e$ and Jacobian $\boldsymbol{J}$ are given (for a given configuration $\boldsymbol{q}$), it is desired to find the solutions $\dot{\boldsymbol{q}}$ that satisfy the linear equation in (3.45) and *minimize* the quadratic cost functional of joint velocities[3]

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{W}\dot{\boldsymbol{q}}$$

where $\boldsymbol{W}$ is a suitable $(n \times n)$ symmetric positive definite weighting matrix.

This problem can be solved with the *method of Lagrange multipliers*. Consider the modified cost functional

$$g(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{W}\dot{\boldsymbol{q}} + \boldsymbol{\lambda}^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}),$$

where $\boldsymbol{\lambda}$ is an $(r \times 1)$ vector of unknown multipliers that allows the incorporation of the constraint (3.45) in the functional to minimize. The requested solution has to satisfy the necessary conditions:

$$\left(\frac{\partial g}{\partial \dot{\boldsymbol{q}}}\right)^T = \boldsymbol{0} \qquad \left(\frac{\partial g}{\partial \boldsymbol{\lambda}}\right)^T = \boldsymbol{0}.$$

From the first one, it is $\boldsymbol{W}\dot{\boldsymbol{q}} - \boldsymbol{J}^T\boldsymbol{\lambda} = \boldsymbol{0}$ and thus

$$\dot{\boldsymbol{q}} = \boldsymbol{W}^{-1}\boldsymbol{J}^T\boldsymbol{\lambda} \tag{3.49}$$

where the inverse of $\boldsymbol{W}$ exists. Notice that the solution (3.49) is a minimum, since $\partial^2 g/\partial \dot{\boldsymbol{q}}^2 = \boldsymbol{W}$ is positive definite. From the second condition above, the constraint

$$\boldsymbol{v}_e = \boldsymbol{J}\dot{\boldsymbol{q}}$$

is recovered. Combining the two conditions gives

$$\boldsymbol{v}_e = \boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T\boldsymbol{\lambda};$$

under the assumption that $\boldsymbol{J}$ has full rank, $\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T$ is an $(r \times r)$ square matrix of rank r and thus can be inverted. Solving for $\boldsymbol{\lambda}$ yields

$$\boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{v}_e$$

[3] Quadratic forms and the relative operations are recalled in Sect. A.6.

which, substituted into (3.49), gives the sought optimal solution

$$\dot{\boldsymbol{q}} = \boldsymbol{W}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{v}_e. \tag{3.50}$$

Premultiplying both sides of (3.50) by $\boldsymbol{J}$, it is easy to verify that this solution satisfies the differential kinematics equation in (3.45).

A particular case occurs when the weighting matrix $\boldsymbol{W}$ is the identity matrix $\boldsymbol{I}$ and the solution simplifies into

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger \boldsymbol{v}_e; \tag{3.51}$$

the matrix

$$\boldsymbol{J}^\dagger = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1} \tag{3.52}$$

is the *right pseudo-inverse* of $\boldsymbol{J}$.[4] The obtained solution locally minimizes the norm of joint velocities.

It was pointed out above that if $\dot{\boldsymbol{q}}^*$ is a solution to (3.45), $\dot{\boldsymbol{q}}^* + \boldsymbol{P}\dot{\boldsymbol{q}}_0$ is also a solution, where $\dot{\boldsymbol{q}}_0$ is a vector of arbitrary joint velocities and $\boldsymbol{P}$ is a projector in the null space of $\boldsymbol{J}$. Therefore, in view of the presence of redundant DOFs, the solution (3.51) can be modified by the introduction of another term of the kind $\boldsymbol{P}\dot{\boldsymbol{q}}_0$. In particular, $\dot{\boldsymbol{q}}_0$ can be specified so as to satisfy an additional constraint to the problem.

In that case, it is necessary to consider a new cost functional in the form

$$g'(\dot{\boldsymbol{q}}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0);$$

this choice is aimed at minimizing the norm of vector $\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0$; in other words, solutions are sought which satisfy the constraint (3.45) and are as close as possible to $\dot{\boldsymbol{q}}_0$. In this way, the objective specified through $\dot{\boldsymbol{q}}_0$ becomes unavoidably a secondary objective to satisfy with respect to the primary objective specified by the constraint (3.45).

Proceeding in a way similar to the above yields

$$g'(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) + \boldsymbol{\lambda}^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}});$$

from the first necessary condition it is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^T\boldsymbol{\lambda} + \dot{\boldsymbol{q}}_0 \tag{3.53}$$

which, substituted into (3.45), gives

$$\boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{J}^T)^{-1}(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}_0).$$

Finally, substituting $\boldsymbol{\lambda}$ back in (3.53) gives

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger \boldsymbol{v}_e + (\boldsymbol{I}_n - \boldsymbol{J}^\dagger \boldsymbol{J})\dot{\boldsymbol{q}}_0. \tag{3.54}$$

[4] See Sect. A.7 for the definition of the pseudo-inverse of a matrix.

As can be easily recognized, the obtained solution is composed of two terms. The first is relative to minimum norm joint velocities. The second, termed *homogeneous solution*, attempts to satisfy the additional constraint to specify via $\dot{\boldsymbol{q}}_0$;[5] the matrix $(\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})$ is one of those matrices $\boldsymbol{P}$ introduced in (3.46) which allows the projection of the vector $\dot{\boldsymbol{q}}_0$ in the null space of $\boldsymbol{J}$, so as not to violate the constraint (3.45). A direct consequence is that, in the case $\boldsymbol{v}_e = \boldsymbol{0}$, is is possible to generate *internal motions* described by $(\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})\dot{\boldsymbol{q}}_0$ that reconfigure the manipulator structure without changing the end-effector position and orientation.

Finally, it is worth discussing the way to specify the vector $\dot{\boldsymbol{q}}_0$ for a convenient utilization of redundant DOFs. A typical choice is

$$\dot{\boldsymbol{q}}_0 = k_0 \left(\frac{\partial w(\boldsymbol{q})}{\partial \boldsymbol{q}} \right)^T \tag{3.55}$$

where $k_0 > 0$ and $w(\boldsymbol{q})$ is a (secondary) objective function of the joint variables. Since the solution moves along the direction of the gradient of the objective function, it attempts to *maximize* it *locally* compatible to the primary objective (kinematic constraint). Typical objective functions are:

- The *manipulability measure*, defined as

$$w(\boldsymbol{q}) = \sqrt{\det\big(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\big)} \tag{3.56}$$

  which vanishes at a singular configuration; thus, by maximizing this measure, redundancy is exploited to move away from singularities.[6]
- The *distance from mechanical joint limits*, defined as

$$w(\boldsymbol{q}) = -\frac{1}{2n} \sum_{i=1}^{n} \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \tag{3.57}$$

  where q_{iM} (q_{im}) denotes the maximum (minimum) joint limit and $\bar{q}_i$ the middle value of the joint range; thus, by maximizing this distance, redundancy is exploited to keep the joint variables as close as possible to the centre of their ranges.
- The *distance from an obstacle*, defined as

$$w(\boldsymbol{q}) = \min_{\boldsymbol{p},\boldsymbol{o}} \|\boldsymbol{p}(\boldsymbol{q}) - \boldsymbol{o}\| \tag{3.58}$$

  where $\boldsymbol{o}$ is the position vector of a suitable point on the obstacle (its centre, for instance, if the obstacle is modelled as a sphere) and $\boldsymbol{p}$ is the

[5] It should be recalled that the additional constraint has secondary priority with respect to the primary kinematic constraint.

[6] The manipulability measure is given by the product of the singular values of the Jacobian (see Problem 3.8).

position vector of a generic point along the structure; thus, by maximizing this distance, redundancy is exploited to avoid collision of the manipulator with an obstacle (see also Problem 3.9).[7]

3.5.2 Kinematic Singularities

Both solutions (3.47) and (3.51) can be computed only when the Jacobian has full rank. Hence, they become meaningless when the manipulator is at a singular configuration; in such a case, the system $\boldsymbol{v}_e = \boldsymbol{J}\dot{\boldsymbol{q}}$ contains linearly dependent equations.

It is possible to find a solution $\dot{\boldsymbol{q}}$ by extracting all the linearly independent equations only if $\boldsymbol{v}_e \in \mathcal{R}(\boldsymbol{J})$. The occurrence of this situation means that the assigned path is physically executable by the manipulator, even though it is at a singular configuration. If instead $\boldsymbol{v}_e \notin \mathcal{R}(\boldsymbol{J})$, the system of equations has no solution; this means that the operational space path cannot be executed by the manipulator at the given posture.

It is important to underline that the inversion of the Jacobian can represent a serious inconvenience not only at a singularity but also in the neighbourhood of a singularity. For instance, for the Jacobian inverse it is well known that its computation requires the computation of the determinant; in the neighbourhood of a singularity, the determinant takes on a relatively small value which can cause large joint velocities (see point **c)** in Sect. 3.3). Consider again the above example of the shoulder singularity for the anthropomorphic arm. If a path is assigned to the end-effector which passes nearby the base rotation axis (geometric locus of singular configurations), the base joint is forced to make a rotation of about π in a relatively short time to allow the end-effector to keep tracking the imposed trajectory.

A more rigorous analysis of the solution features in the neighbourhood of singular configurations can be developed by resorting to the singular value decomposition (SVD) of matrix $\boldsymbol{J}$.[8]

An alternative solution overcoming the problem of inverting differential kinematics in the neighbourhood of a singularity is provided by the so-called *damped least-squares (DLS) inverse*

$$\boldsymbol{J}^{\star} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T + k^2\boldsymbol{I})^{-1} \tag{3.59}$$

where k is a damping factor that renders the inversion better conditioned from a numerical viewpoint. It can be shown that such a solution can be

[7] If an obstacle occurs along the end-effector path, it is opportune to invert the order of priority between the kinematic constraint and the additional constraint; in this way the obstacle may be avoided, but one gives up tracking the desired path.

[8] See Sect. A.8.

obtained by reformulating the problem in terms of the minimization of the cost functional

$$g''(\dot{\boldsymbol{q}}) = \frac{1}{2}(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}})^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}) + \frac{1}{2}k^2\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}},$$

where the introduction of the first term allows a finite inversion error to be tolerated, with the advantage of norm-bounded velocities. The factor k establishes the relative weight between the two objectives, and there exist techniques for selecting optimal values for the damping factor (see Problem 3.10).

3.6 Analytical Jacobian

The above sections have shown the way to compute the end-effector velocity in terms of the velocity of the end-effector frame. The Jacobian is computed according to a *geometric technique* in which the contributions of each joint velocity to the components of end-effector linear and angular velocity are determined.

If the end-effector pose is specified in terms of a minimal number of parameters in the operational space as in (2.80), it is natural to ask whether it is possible to compute the Jacobian via differentiation of the direct kinematics function with respect to the joint variables. To this end, an *analytical technique* is presented below to compute the Jacobian, and the existing relationship between the two Jacobians is found.

The translational velocity of the end-effector frame can be expressed as the time derivative of vector $\boldsymbol{p}_e$, representing the origin of the end-effector frame with respect to the base frame, i.e.,

$$\dot{\boldsymbol{p}}_e = \frac{\partial \boldsymbol{p}_e}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.60}$$

For what concerns the rotational velocity of the end-effector frame, the minimal representation of orientation in terms of three variables $\boldsymbol{\phi}_e$ can be considered. Its time derivative $\dot{\boldsymbol{\phi}}_e$ in general differs from the angular velocity vector defined above. In any case, once the function $\boldsymbol{\phi}_e(\boldsymbol{q})$ is known, it is formally correct to consider the Jacobian obtained as

$$\dot{\boldsymbol{\phi}}_e = \frac{\partial \boldsymbol{\phi}_e}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{J}_\phi(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.61}$$

Computing the Jacobian $\boldsymbol{J}_\phi(\boldsymbol{q})$ as $\partial\boldsymbol{\phi}_e/\partial\boldsymbol{q}$ is not straightforward, since the function $\boldsymbol{\phi}_e(\boldsymbol{q})$ is not usually available in direct form, but requires computation of the elements of the relative rotation matrix.

Upon these premises, the differential kinematics equation can be obtained as the time derivative of the direct kinematics equation in (2.82), i.e.,

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_\phi(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.62}$$

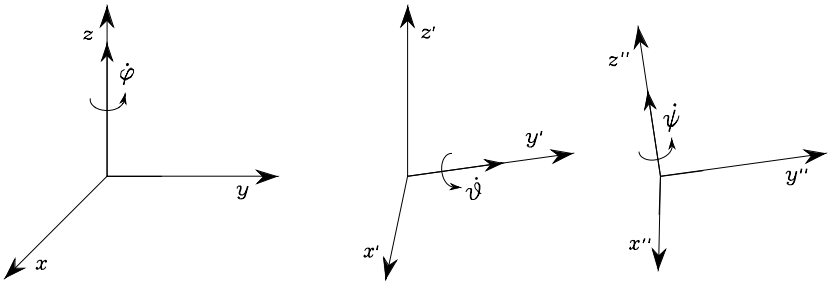

Fig. 3.8. Rotational velocities of Euler angles ZYZ in current frame

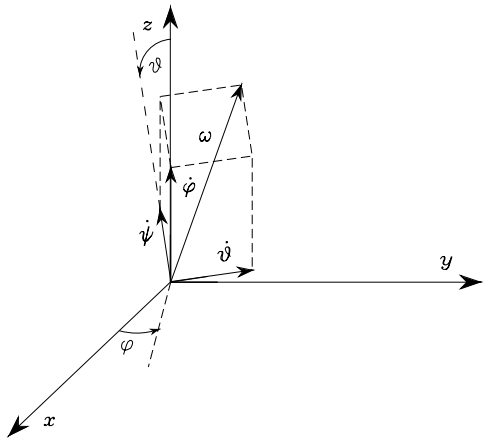

Fig. 3.9. Composition of elementary rotational velocities for computing angular velocity

where the *analytical Jacobian*

$$\boldsymbol{J}_A(\boldsymbol{q}) = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{3.63}$$

is different from the geometric Jacobian $\boldsymbol{J}$, since the end-effector angular velocity $\boldsymbol{\omega}_e$ with respect to the base frame is not given by $\dot{\boldsymbol{\phi}}_e$.

It is possible to find the relationship between the angular velocity $\boldsymbol{\omega}_e$ and the rotational velocity $\dot{\boldsymbol{\phi}}_e$ for a given set of orientation angles. For instance, consider the Euler angles ZYZ defined in Sect. 2.4.1; in Fig. 3.8, the vectors corresponding to the rotational velocities $\dot{\varphi}$, $\dot{\vartheta}$, $\dot{\psi}$ have been represented with reference to the current frame. Figure 3.9 illustrates how to compute the contributions of each rotational velocity to the components of angular velocity about the axes of the reference frame:

- as a result of $\dot{\varphi}$: $[\,\omega_x \quad \omega_y \quad \omega_z\,]^T = \dot{\varphi}\,[\,0 \quad 0 \quad 1\,]^T$
- as a result of $\dot{\vartheta}$: $[\,\omega_x \quad \omega_y \quad \omega_z\,]^T = \dot{\vartheta}\,[\,-s_\varphi \quad c_\varphi \quad 0\,]^T$

- as a result of $\dot{\psi}$: $[\,\omega_x \quad \omega_y \quad \omega_z\,]^T = \dot{\psi}\,[\,c_\varphi s_\vartheta \quad s_\varphi s_\vartheta \quad c_\vartheta\,]^T$,

and then the equation relating the angular velocity $\boldsymbol{\omega}_e$ to the time derivative of the Euler angles $\dot{\boldsymbol{\phi}}_e$ is[9]

$$\boldsymbol{\omega}_e = \boldsymbol{T}(\boldsymbol{\phi}_e)\dot{\boldsymbol{\phi}}_e, \tag{3.64}$$

where, in this case,

$$\boldsymbol{T} = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\vartheta \\ 0 & c_\varphi & s_\varphi s_\vartheta \\ 1 & 0 & c_\vartheta \end{bmatrix}.$$

The determinant of matrix $\boldsymbol{T}$ is $-s_\vartheta$, which implies that the relationship cannot be inverted for $\vartheta = 0, \pi$. This means that, even though all rotational velocities of the end-effector frame can be expressed by means of a suitable angular velocity vector $\boldsymbol{\omega}_e$, there exist angular velocities which cannot be expressed by means of $\dot{\boldsymbol{\phi}}_e$ when the orientation of the end-effector frame causes $s_\vartheta = 0$.[10] In fact, in this situation, the angular velocities that can be described by $\dot{\boldsymbol{\phi}}_e$ should have linearly dependent components in the directions orthogonal to axis z ($\omega_x^2 + \omega_y^2 = \dot{\vartheta}^2$). An orientation for which the determinant of the transformation matrix vanishes is termed *representation singularity* of $\boldsymbol{\phi}_e$.

From a physical viewpoint, the meaning of $\boldsymbol{\omega}_e$ is more intuitive than that of $\dot{\boldsymbol{\phi}}_e$. The three components of $\boldsymbol{\omega}_e$ represent the components of angular velocity with respect to the base frame. Instead, the three elements of $\dot{\boldsymbol{\phi}}_e$ represent nonorthogonal components of angular velocity defined with respect to the axes of a frame that varies as the end-effector orientation varies. On the other hand, while the integral of $\dot{\boldsymbol{\phi}}_e$ over time gives $\boldsymbol{\phi}_e$, the integral of $\boldsymbol{\omega}_e$ does not admit a clear physical interpretation, as can be seen in the following example.

Example 3.3

Consider an object whose orientation with respect to a reference frame is known at time $t = 0$. Assign the following time profiles to $\boldsymbol{\omega}$:

- $\boldsymbol{\omega} = [\,\pi/2 \quad 0 \quad 0\,]^T \quad 0 \le t \le 1 \qquad \boldsymbol{\omega} = [\,0 \quad \pi/2 \quad 0\,]^T \quad 1 < t \le 2,$
- $\boldsymbol{\omega} = [\,0 \quad \pi/2 \quad 0\,]^T \quad 0 \le t \le 1 \qquad \boldsymbol{\omega} = [\,\pi/2 \quad 0 \quad 0\,]^T \quad 1 < t \le 2.$

The integral of $\boldsymbol{\omega}$ gives the same result in the two cases

$$\int_0^2 \boldsymbol{\omega} dt = [\,\pi/2 \quad \pi/2 \quad 0\,]^T$$

but the final object orientation corresponding to the second timing law is clearly different from the one obtained with the first timing law (Fig. 3.10).

---

[9] This relation can also be obtained from the rotation matrix associated with the three angles (see Problem 3.11).

[10] In Sect. 2.4.1, it was shown that for this orientation the inverse solution of the Euler angles degenerates.

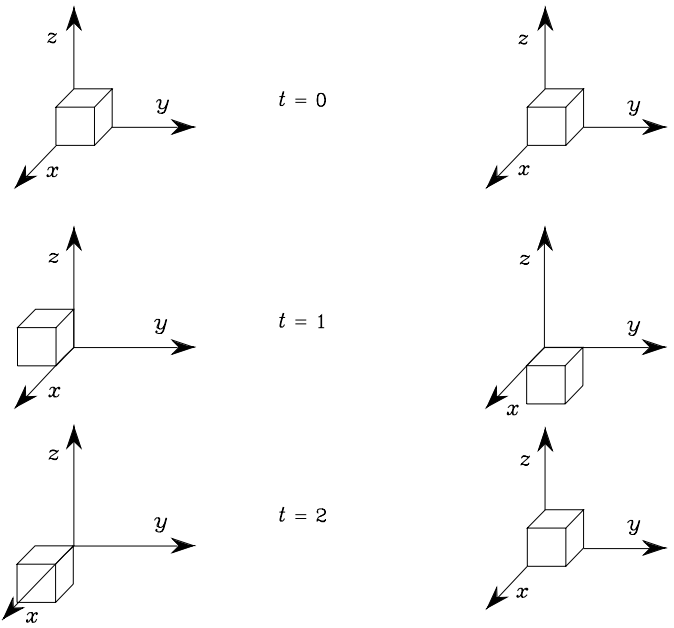

Fig. 3.10. Nonuniqueness of orientation computed as the integral of angular velocity

Once the transformation $\boldsymbol{T}$ between $\boldsymbol{\omega}_e$ and $\dot{\boldsymbol{\phi}}_e$ is given, the analytical Jacobian can be related to the geometric Jacobian as

$$\boldsymbol{v}_e = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}(\boldsymbol{\phi}_e) \end{bmatrix} \dot{\boldsymbol{x}}_e = \boldsymbol{T}_A(\boldsymbol{\phi}_e)\dot{\boldsymbol{x}}_e \tag{3.65}$$

which, in view of (3.4), (3.62), yields

$$\boldsymbol{J} = \boldsymbol{T}_A(\boldsymbol{\phi})\boldsymbol{J}_A. \tag{3.66}$$

This relationship shows that $\boldsymbol{J}$ and $\boldsymbol{J}_A$, in general, differ. Regarding the use of either one or the other in all those problems where the influence of the Jacobian matters, it is anticipated that the geometric Jacobian will be adopted whenever it is necessary to refer to quantities of clear physical meaning, while the analytical Jacobian will be adopted whenever it is necessary to refer to differential quantities of variables defined in the operational space.

For certain manipulator geometries, it is possible to establish a substantial equivalence between $\boldsymbol{J}$ and $\boldsymbol{J}_A$. In fact, when the DOFs cause rotations of the end-effector all about the same fixed axis in space, the two Jacobians are essentially the same. This is the case of the above three-link planar arm. Its geometric Jacobian (3.35) reveals that only rotations about axis z_0 are permitted. The (3×3) analytical Jacobian that can be derived by considering the end-effector position components in the plane of the structure and defining

the end-effector orientation as $\phi = \vartheta_1 + \vartheta_2 + \vartheta_3$ coincides with the matrix that is obtained by eliminating the three null rows of the geometric Jacobian.

3.7 Inverse Kinematics Algorithms

In Sect. 3.5 it was shown how to invert kinematics by using the differential kinematics equation. In the numerical implementation of (3.48), computation of joint velocities is obtained by using the inverse of the Jacobian evaluated with the joint variables at the previous instant of time

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{q}(t_k) + \boldsymbol{J}^{-1}(\boldsymbol{q}(t_k))\boldsymbol{v}_e(t_k)\Delta t.$$

It follows that the computed joint velocities $\dot{\boldsymbol{q}}$ do not coincide with those satisfying (3.47) in the continuous time. Therefore, reconstruction of joint variables $\boldsymbol{q}$ is entrusted to a numerical integration which involves *drift* phenomena of the solution; as a consequence, the end-effector pose corresponding to the computed joint variables differs from the desired one.

This inconvenience can be overcome by resorting to a solution scheme that accounts for the *operational space error* between the desired and the actual end-effector position and orientation. Let

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}_e \tag{3.67}$$

be the expression of such error.

Consider the time derivative of (3.67), i.e.,

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_e \tag{3.68}$$

which, according to differential kinematics (3.62), can be written as

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.69}$$

Notice in (3.69) that the use of operational space quantities has naturally lead to using the analytical Jacobian in lieu of the geometric Jacobian. For this equation to lead to an *inverse kinematics algorithm*, it is worth relating the computed joint velocity vector $\dot{\boldsymbol{q}}$ to the error $\boldsymbol{e}$ so that (3.69) gives a differential equation describing error evolution over time. Nonetheless, it is necessary to choose a relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{e}$ that ensures convergence of the error to zero.

Having formulated inverse kinematics in algorithmic terms implies that the joint variables $\boldsymbol{q}$ corresponding to a given end-effector pose $\boldsymbol{x}_d$ are accurately computed only when the error $\boldsymbol{x}_d - \boldsymbol{k}(\boldsymbol{q})$ is reduced within a given threshold; such settling time depends on the dynamic characteristics of the error differential equation. The choice of $\dot{\boldsymbol{q}}$ as a function of $\boldsymbol{e}$ permits finding inverse kinematics algorithms with different features.

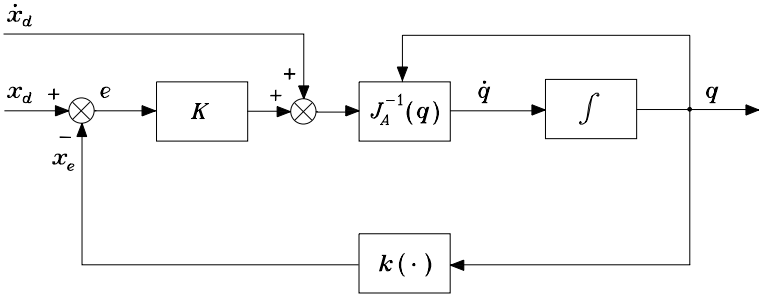

Fig. 3.11. Inverse kinematics algorithm with Jacobian inverse

3.7.1 Jacobian (Pseudo-)inverse

On the assumption that matrix $\boldsymbol{J}_A$ is square and nonsingular, the choice

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e}) \tag{3.70}$$

leads to the equivalent linear system

$$\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = \boldsymbol{0}. \tag{3.71}$$

If $\boldsymbol{K}$ is a positive definite (usually diagonal) matrix, the system (3.71) is *asymptotically stable*. The error tends to zero along the trajectory with a convergence rate that depends on the eigenvalues of matrix $\boldsymbol{K}$;[11] the larger the eigenvalues, the faster the convergence. Since the scheme is practically implemented as a discrete-time system, it is reasonable to predict that an upper bound exists on the eigenvalues; depending on the sampling time, there will be a limit for the maximum eigenvalue of $\boldsymbol{K}$ under which asymptotic stability of the error system is guaranteed.

The block scheme corresponding to the inverse kinematics algorithm in (3.70) is illustrated in Fig. 3.11, where $\boldsymbol{k}(\cdot)$ indicates the direct kinematics function in (2.82). This scheme can be revisited in terms of the usual feedback control schemes. Specifically, it can observed that the nonlinear block $\boldsymbol{k}(\cdot)$ is needed to compute $\boldsymbol{x}$ and thus the tracking error $\boldsymbol{e}$, while the block $\boldsymbol{J}_A^{-1}(\boldsymbol{q})$ has been introduced to compensate for $\boldsymbol{J}_A(\boldsymbol{q})$ and making the system linear. The block scheme shows the presence of a string of integrators on the forward loop and then, for a constant reference ($\dot{\boldsymbol{x}}_d = \boldsymbol{0}$), guarantees a null steady-state error. Further, the *feedforward* action provided by $\dot{\boldsymbol{x}}_d$ for a time-varying reference ensures that the error is kept to zero (in the case $\boldsymbol{e}(0) = \boldsymbol{0}$) along the whole trajectory, independently of the type of desired reference $\boldsymbol{x}_d(t)$.

Finally, notice that (3.70), for $\dot{\boldsymbol{x}}_d = \boldsymbol{0}$, corresponds to the Newton method for solving a system of nonlinear equations. Given a constant end-effector pose $\boldsymbol{x}_d$, the algorithm can be keenly applied to compute one of the admissible

[11] See Sect. A.5.

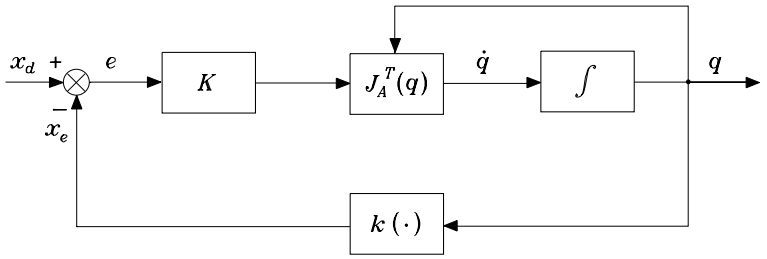

Fig. 3.12. Block scheme of the inverse kinematics algorithm with Jacobian transpose

solutions to the inverse kinematics problem, whenever that does not admit closed-form solutions, as discussed in Sect. 2.12. Such a method is also useful in practice at the start-up of the manipulator for a given task, to compute the corresponding joint configuration.

In the case of a *redundant manipulator*, solution (3.70) can be generalized into

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\dagger}(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e}) + (\boldsymbol{I}_n - \boldsymbol{J}_A^{\dagger}\boldsymbol{J}_A)\dot{\boldsymbol{q}}_0, \tag{3.72}$$

which represents the algorithmic version of solution (3.54).

The structure of the inverse kinematics algorithm can be conceptually adopted for a simple robot control technique, known under the name of *kinematic control*. As will be seen in Chap. 7, a manipulator is actually an electromechanical system actuated by motor torques, while in Chaps. 8–10 dynamic control techniques will be presented which will properly account for the nonlinear and coupling effects of the dynamic model.

At first approximation, however, it is possible to consider a kinematic command as system input, typically a velocity. This is possible in view of the presence of a low-level control loop, which 'ideally' imposes any specified reference velocity. On the other hand, such a loop already exists in a 'closed' control unit, where the user can also intervene with kinematic commands. In other words, the scheme in Fig. 3.11 can implement a kinematic control, provided that the integrator is regarded as a simplified model of the robot, thanks to the presence of single joint local servos, which ensure a more or less accurate reproduction of the velocity commands. Nevertheless, it is worth underlining that such a kinematic control technique yields satisfactory performance only when one does not require too fast motions or rapid accelerations. The performance of the independent joint control will be analyzed in Sect. 8.3.

3.7.2 Jacobian Transpose

A computationally simpler algorithm can be derived by finding a relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{e}$ that ensures error convergence to zero, without requiring linearization of (3.69). As a consequence, the error dynamics is governed by a

nonlinear differential equation. The Lyapunov direct method can be utilized to determine a dependence $\dot{\boldsymbol{q}}(\boldsymbol{e})$ that ensures asymptotic stability of the error system. Choose as Lyapunov function candidate the positive definite quadratic form[12]

$$V(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K}\boldsymbol{e}, \tag{3.73}$$

where $\boldsymbol{K}$ is a symmetric positive definite matrix. This function is so that

$$V(\boldsymbol{e}) > 0 \quad \forall \boldsymbol{e} \neq \boldsymbol{0}, \qquad V(\boldsymbol{0}) = 0.$$

Differentiating (3.73) with respect to time and accounting for (3.68) gives

$$\dot{V} = \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_e. \tag{3.74}$$

In view of (3.62), it is

$$\dot{V} = \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.75}$$

At this point, the choice of joint velocities as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e} \tag{3.76}$$

leads to

$$\dot{V} = \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e}. \tag{3.77}$$

Consider the case of a constant reference ($\dot{\boldsymbol{x}}_d = \boldsymbol{0}$). The function in (3.77) is negative definite, under the assumption of full rank for $\boldsymbol{J}_A(\boldsymbol{q})$. The condition $\dot{V} < 0$ with $V > 0$ implies that the system trajectories uniformly converge to $\boldsymbol{e} = \boldsymbol{0}$, i.e., the system is *asymptotically stable*. When $\mathcal{N}(\boldsymbol{J}_A^T) \neq \emptyset$, the function in (3.77) is only negative semi-definite, since $\dot{V} = 0$ for $\boldsymbol{e} \neq \boldsymbol{0}$ with $\boldsymbol{K}\boldsymbol{e} \in \mathcal{N}(\boldsymbol{J}_A^T)$. In this case, the algorithm can get stuck at $\dot{\boldsymbol{q}} = \boldsymbol{0}$ with $\boldsymbol{e} \neq \boldsymbol{0}$. However, the example that follows will show that this situation occurs only if the assigned end-effector position is not actually reachable from the current configuration.

The resulting block scheme is illustrated in Fig. 3.12, which shows the notable feature of the algorithm to require *computation only of direct kinematics functions* $\boldsymbol{k}(\boldsymbol{q})$, $\boldsymbol{J}_A^T(\boldsymbol{q})$.

It can be recognized that (3.76) corresponds to the gradient method for the solution of a system on nonlinear equations. As in the case of the Jacobian inverse solution, for a given constant end-effector pose $\boldsymbol{x}_d$, the Jacobian transpose algorithm can be keenly employed to solve the inverse kinematics problem, or more simply to initialize the values of the manipulator joint variables.

The case when $\boldsymbol{x}_d$ is a time-varying function ($\dot{\boldsymbol{x}}_d \neq \boldsymbol{0}$) deserves a separate analysis. In order to obtain $\dot{V} < 0$ also in this case, it would be sufficient to choose a $\dot{\boldsymbol{q}}$ that depends on the (pseudo-)inverse of the Jacobian as in (3.70),

[12] See Sect. C.3 for the presentation of the Lyapunov direct method.

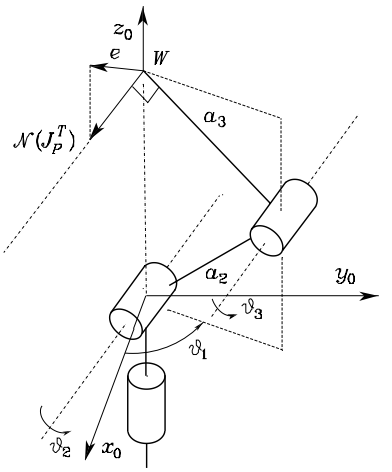

Fig. 3.13. Characterization of the anthropomorphic arm at a shoulder singularity for the admissible solutions of the Jacobian transpose algorithm

recovering the asymptotic stability result derived above.[13] For the inversion scheme based on the transpose, the first term on the right-hand side of (3.77) is not cancelled any more and nothing can be said about its sign. This implies that asymptotic stability along the trajectory cannot be achieved. The tracking error $\boldsymbol{e}(t)$ is, anyhow, norm-bounded; the larger the norm of $\boldsymbol{K}$, the smaller the norm of $\boldsymbol{e}$.[14] In practice, since the inversion scheme is to be implemented in discrete-time, there is an upper bound on the norm of $\boldsymbol{K}$ with reference to the adopted sampling time.

Example 3.4

Consider the anthropomorphic arm; a shoulder singularity occurs whenever $a_2c_2 + a_3c_{23} = 0$ (Fig. 3.6). In this configuration, the transpose of the Jacobian in (3.38) is

$$\boldsymbol{J}_P^T = \begin{bmatrix} 0 & 0 & 0 \\ -c_1(a_2s_2 + a_3s_{23}) & -s_1(a_2s_2 + a_3s_{23}) & 0 \\ -a_3c_1s_{23} & -a_3s_1s_{23} & a_3c_{23} \end{bmatrix}.$$

By computing the null space of $\boldsymbol{J}_P^T$, if ν_x, ν_y and ν_z denote the components of vector $\boldsymbol{\nu}$ along the axes of the base frame, one has the result

$$\frac{\nu_y}{\nu_x} = -\frac{1}{\tan\vartheta_1} \qquad \nu_z = 0,$$

[13] Notice that, anyhow, in case of kinematic singularities, it is necessary to resort to an inverse kinematics scheme that does not require inversion of the Jacobian.

[14] Notice that the negative definite term is a quadratic function of the error, while the other term is a linear function of the error. Therefore, for an error of very small norm, the linear term prevails over the quadratic term, and the norm of $\boldsymbol{K}$ should be increased to reduce the norm of $\boldsymbol{e}$ as much as possible.

implying that the direction of $\mathcal{N}(\boldsymbol{J}_P^T)$ coincides with the direction orthogonal to the plane of the structure (Fig. 3.13). The Jacobian transpose algorithm gets stuck if, with $\boldsymbol{K}$ diagonal and having all equal elements, the desired position is along the line normal to the plane of the structure at the intersection with the wrist point. On the other hand, the end-effector cannot physically move from the singular configuration along such a line. Instead, if the prescribed path has a non-null component in the plane of the structure at the singularity, algorithm convergence is ensured, since in that case $\boldsymbol{K}\boldsymbol{e} \notin \mathcal{N}(\boldsymbol{J}_P^T)$.

In summary, the algorithm based on the computation of the Jacobian transpose provides a computationally efficient inverse kinematics method that can be utilized also for paths crossing kinematic singularities.

3.7.3 Orientation Error

The inverse kinematics algorithms presented in the above sections utilize the analytical Jacobian since they operate on error variables (position and orientation) that are defined in the operational space.

For what concerns the position error, it is obvious that its expression is given by

$$\boldsymbol{e}_P = \boldsymbol{p}_d - \boldsymbol{p}_e(\boldsymbol{q}) \tag{3.78}$$

where $\boldsymbol{p}_d$ and $\boldsymbol{p}_e$ denote respectively the desired and computed end-effector positions. Further, its time derivative is

$$\dot{\boldsymbol{e}}_P = \dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}_e. \tag{3.79}$$

On the other hand, for what concerns the *orientation error*, its expression depends on the particular representation of end-effector orientation, namely, Euler angles, angle and axis, and unit quaternion.

Euler angles

The orientation error is chosen according to an expression formally analogous to (3.78), i.e.,

$$\boldsymbol{e}_O = \boldsymbol{\phi}_d - \boldsymbol{\phi}_e(\boldsymbol{q}) \tag{3.80}$$

where $\boldsymbol{\phi}_d$ and $\boldsymbol{\phi}_e$ denote respectively the desired and computed set of Euler angles. Further, its time derivative is

$$\dot{\boldsymbol{e}}_O = \dot{\boldsymbol{\phi}}_d - \dot{\boldsymbol{\phi}}_e. \tag{3.81}$$

Therefore, assuming that neither kinematic nor representation singularities occur, the Jacobian inverse solution for a nonredundant manipulator is derived from (3.70), i.e.,

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P \boldsymbol{e}_P \\ \dot{\boldsymbol{\phi}}_d + \boldsymbol{K}_O \boldsymbol{e}_O \end{bmatrix} \tag{3.82}$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_O$ are positive definite matrices.

As already pointed out in Sect. 2.10 for computation of the direct kinematics function in the form (2.82), the determination of the orientation variables from the joint variables is not easy except for simple cases (see Example 2.5). To this end, it is worth recalling that computation of the angles $\boldsymbol{\phi}_e$, in a minimal representation of orientation, requires computation of the rotation matrix $\boldsymbol{R}_e = [\, \boldsymbol{n}_e \quad \boldsymbol{s}_e \quad \boldsymbol{a}_e \,]$; in fact, only the dependence of $\boldsymbol{R}_e$ on $\boldsymbol{q}$ is known in closed form, but not that of $\boldsymbol{\phi}_e$ on $\boldsymbol{q}$. Further, the use of inverse functions (Atan2) in (2.19), (2.22) involves a non-negligible complexity in the computation of the analytical Jacobian, and the occurrence of representation singularities constitutes another drawback for the orientation error based on Euler angles.

Different kinds of remarks are to be made about the way to assign a time profile for the reference variables $\boldsymbol{\phi}_d$ chosen to represent end-effector orientation. The most intuitive way to specify end-effector orientation is to refer to the orientation of the end-effector frame ($\boldsymbol{n}_d$, $\boldsymbol{s}_d$, $\boldsymbol{a}_d$) with respect to the base frame. Given the limitations pointed out in Sect. 2.10 about guaranteeing orthonormality of the unit vectors along time, it is necessary first to compute the Euler angles corresponding to the initial and final orientation of the end-effector frame via (2.19), (2.22); only then a time evolution can be generated. Such solutions will be presented in Chap. 4.

A radical simplification of the problem at issue can be obtained for manipulators having a spherical wrist. Section 2.12.2 pointed out the possibility to solve the inverse kinematics problem for the position part separately from that for the orientation part. This result also has an impact at algorithmic level. In fact, the implementation of an inverse kinematics algorithm for determining the joint variables influencing the wrist position allows the computation of the time evolution of the wrist frame $\boldsymbol{R}_W(t)$. Hence, once the desired time evolution of the end-effector frame $\boldsymbol{R}_d(t)$ is given, it is sufficient to compute the Euler angles ZYZ from the matrix $\boldsymbol{R}_W^T \boldsymbol{R}_d$ by applying (2.19). As shown in Sect. 2.12.5, these angles are directly the joint variables of the spherical wrist. See also Problem 3.14.

The above considerations show that the inverse kinematics algorithms based on the analytical Jacobian are effective for kinematic structures having a spherical wrist which are of significant interest. For manipulator structures which cannot be reduced to that class, it may be appropriate to reformulate the inverse kinematics problem on the basis of a different definition of the orientation error.

Angle and axis

If $\boldsymbol{R}_d = [\,\boldsymbol{n}_d \quad \boldsymbol{s}_d \quad \boldsymbol{a}_d\,]$ denotes the desired rotation matrix of the end-effector frame and $\boldsymbol{R}_e = [\,\boldsymbol{n}_e \quad \boldsymbol{s}_e \quad \boldsymbol{a}_e\,]$ the rotation matrix that can be computed from the joint variables, the orientation error between the two frames can be expressed as

$$\boldsymbol{e}_O = \boldsymbol{r} \sin \vartheta \tag{3.83}$$

where ϑ and $\boldsymbol{r}$ identify the *angle and axis* of the equivalent rotation that can be deduced from the matrix

$$\boldsymbol{R}(\vartheta, \boldsymbol{r}) = \boldsymbol{R}_d \boldsymbol{R}_e^T(\boldsymbol{q}), \tag{3.84}$$

describing the rotation needed to align $\boldsymbol{R}$ with $\boldsymbol{R}_d$. Notice that (3.83) gives a unique relationship for $-\pi/2 < \vartheta < \pi/2$. The angle ϑ represents the magnitude of an orientation error, and thus the above limitation is not restrictive since the tracking error is typically small for an inverse kinematics algorithm.

By comparing the off-diagonal terms of the expression of $\boldsymbol{R}(\vartheta, \boldsymbol{r})$ in (2.25) with the corresponding terms resulting on the right-hand side of (3.84), it can be found that a functional expression of the orientation error in (3.83) is (see Problem 3.16)

$$\boldsymbol{e}_O = \frac{1}{2}(\boldsymbol{n}_e(\boldsymbol{q}) \times \boldsymbol{n}_d + \boldsymbol{s}_e(\boldsymbol{q}) \times \boldsymbol{s}_d + \boldsymbol{a}_e(\boldsymbol{q}) \times \boldsymbol{a}_d); \tag{3.85}$$

the limitation on ϑ is transformed in the condition $\boldsymbol{n}_e^T \boldsymbol{n}_d \geq 0$, $\boldsymbol{s}_e^T \boldsymbol{s}_d \geq 0$, $\boldsymbol{a}_e^T \boldsymbol{a}_d \geq 0$.

Differentiating (3.85) with respect to time and accounting for the expression of the columns of the derivative of a rotation matrix in (3.8) gives (see Problem 3.19)

$$\dot{\boldsymbol{e}}_O = \boldsymbol{L}^T \boldsymbol{\omega}_d - \boldsymbol{L} \boldsymbol{\omega}_e \tag{3.86}$$

where

$$\boldsymbol{L} = -\frac{1}{2}\big(\boldsymbol{S}(\boldsymbol{n}_d)\boldsymbol{S}(\boldsymbol{n}_e) + \boldsymbol{S}(\boldsymbol{s}_d)\boldsymbol{S}(\boldsymbol{s}_e) + \boldsymbol{S}(\boldsymbol{a}_d)\boldsymbol{S}(\boldsymbol{a}_e)\big). \tag{3.87}$$

At this point, by exploiting the relations (3.2), (3.3) of the geometric Jacobian expressing $\dot{\boldsymbol{p}}_e$ and $\boldsymbol{\omega}_e$ as a function of $\dot{\boldsymbol{q}}$, (3.79), (3.86) become

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \dot{\boldsymbol{e}}_P \\ \dot{\boldsymbol{e}}_O \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_d - \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{L}^T \boldsymbol{\omega}_d - \boldsymbol{L}\boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_d \\ \boldsymbol{L}^T \boldsymbol{\omega}_d \end{bmatrix} - \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{L} \end{bmatrix} \boldsymbol{J}\dot{\boldsymbol{q}}. \tag{3.88}$$

The expression in (3.88) suggests the possibility of devising inverse kinematics algorithms analogous to the ones derived above, but using the geometric Jacobian in place of the analytical Jacobian. For instance, the Jacobian inverse solution for a nonredundant nonsingular manipulator is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P \boldsymbol{e}_P \\ \boldsymbol{L}^{-1} \left(\boldsymbol{L}^T \boldsymbol{\omega}_d + \boldsymbol{K}_O \boldsymbol{e}_O \right) \end{bmatrix}. \tag{3.89}$$

It is worth remarking that the inverse kinematics solution based on (3.89) is expected to perform better than the solution based on (3.82) since it uses the geometric Jacobian in lieu of the analytical Jacobian, thus avoiding the occurrence of representation singularities.

Unit quaternion

In order to devise an inverse kinematics algorithm based on the *unit quaternion*, a suitable orientation error should be defined. Let $\mathcal{Q}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ and $\mathcal{Q}_e = \{\eta_e, \boldsymbol{\epsilon}_e\}$ represent the quaternions associated with $\boldsymbol{R}_d$ and $\boldsymbol{R}_e$, respectively. The orientation error can be described by the rotation matrix $\boldsymbol{R}_d\boldsymbol{R}_e^T$ and, in view of (2.37), can be expressed in terms of the quaternion $\Delta\mathcal{Q} = \{\Delta\eta, \Delta\boldsymbol{\epsilon}\}$ where

$$\Delta\mathcal{Q} = \mathcal{Q}_d * \mathcal{Q}_e^{-1}. \tag{3.90}$$

It can be recognized that $\Delta\mathcal{Q} = \{1, \mathbf{0}\}$ if and only if $\boldsymbol{R}_e$ and $\boldsymbol{R}_d$ are aligned. Hence, it is sufficient to define the orientation error as

$$\boldsymbol{e}_O = \Delta\boldsymbol{\epsilon} = \eta_e(\boldsymbol{q})\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon}_e(\boldsymbol{q}) - \boldsymbol{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}_e(\boldsymbol{q}), \tag{3.91}$$

where the skew-symmetric operator $\boldsymbol{S}(\cdot)$ has been used. Notice, however, that the explicit computation of η_e and $\boldsymbol{\epsilon}_e$ from the joint variables is not possible but it requires the intermediate computation of the rotation matrix $\boldsymbol{R}_e$ that is available from the manipulator direct kinematics; then, the quaternion can be extracted using (2.34).

At this point, a Jacobian inverse solution can be computed as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P\boldsymbol{e}_P \\ \boldsymbol{\omega}_d + \boldsymbol{K}_O\boldsymbol{e}_O \end{bmatrix} \tag{3.92}$$

where noticeably the geometric Jacobian has been used. Substituting (3.92) into (3.4) gives (3.79) and

$$\boldsymbol{\omega}_d - \boldsymbol{\omega}_e + \boldsymbol{K}_O\boldsymbol{e}_O = \mathbf{0}. \tag{3.93}$$

It should be observed that now the orientation error equation is nonlinear in $\boldsymbol{e}_O$ since it contains the end-effector angular velocity error instead of the time derivative of the orientation error. To this end, it is worth considering the relationship between the time derivative of the quaternion $\mathcal{Q}_e$ and the angular velocity $\boldsymbol{\omega}_e$. This can be found to be (see Problem 3.19)

$$\dot{\eta}_e = -\frac{1}{2}\boldsymbol{\epsilon}_e^T\boldsymbol{\omega}_e \tag{3.94}$$

$$\dot{\boldsymbol{\epsilon}}_e = \frac{1}{2}\left(\eta_e\boldsymbol{I}_3 - \boldsymbol{S}(\boldsymbol{\epsilon}_e)\right)\boldsymbol{\omega}_e \tag{3.95}$$

which is the so-called *quaternion propagation*. A similar relationship holds between the time derivative of $\mathcal{Q}_d$ and $\boldsymbol{\omega}_d$.

To study stability of system (3.93), consider the positive definite Lyapunov function candidate

$$V = (\eta_d - \eta_e)^2 + (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_e)^T(\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_e). \tag{3.96}$$

In view of (3.94), (3.95), differentiating (3.96) with respect to time and accounting for (3.93) yields (see Problem 3.20)

$$\dot{V} = -\boldsymbol{e}_O^T \boldsymbol{K}_O \boldsymbol{e}_O \tag{3.97}$$

which is negative definite, implying that $\boldsymbol{e}_O$ converges to zero.

In summary, the inverse kinematics solution based on (3.92) uses the geometric Jacobian as the solution based on (3.89) but is computationally lighter.

3.7.4 Second-order Algorithms

The above inverse kinematics algorithms can be defined as *first-order* algorithms, in that they allow the inversion of a motion trajectory, specified at the end-effector in terms of of position and orientation, into the equivalent joint positions and velocities.

Nevertheless, as will be seen in Chap. 8, for control purposes it may be necessary to invert a motion trajectory specified in terms of position, velocity and acceleration. On the other hand, the manipulator is inherently a *second-order* mechanical system, as will be revealed by the dynamic model to be derived in Chap. 7.

The time differentiation of the differential kinematics equation (3.62) leads to

$$\ddot{\boldsymbol{x}}_e = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \tag{3.98}$$

which gives the relationship between the joint space accelerations and the operational space accelerations.

Under the assumption of a square and non-singular matrix $\boldsymbol{J}_A$, the second-order differential kinematics (3.98) can be inverted in terms of the joint accelerations

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \left(\ddot{\boldsymbol{x}}_e - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \right). \tag{3.99}$$

The numerical integration of (3.99) to reconstruct the joint velocities and positions would unavoidably lead to a drift of the solution; therefore, similarly to the inverse kinematics algorithm with the Jacobian inverse, it is worth considering the error defined in (3.68) along with its derivative

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{x}}_d - \ddot{\boldsymbol{x}}_e \tag{3.100}$$

which, in view of (3.98), yields

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}. \tag{3.101}$$

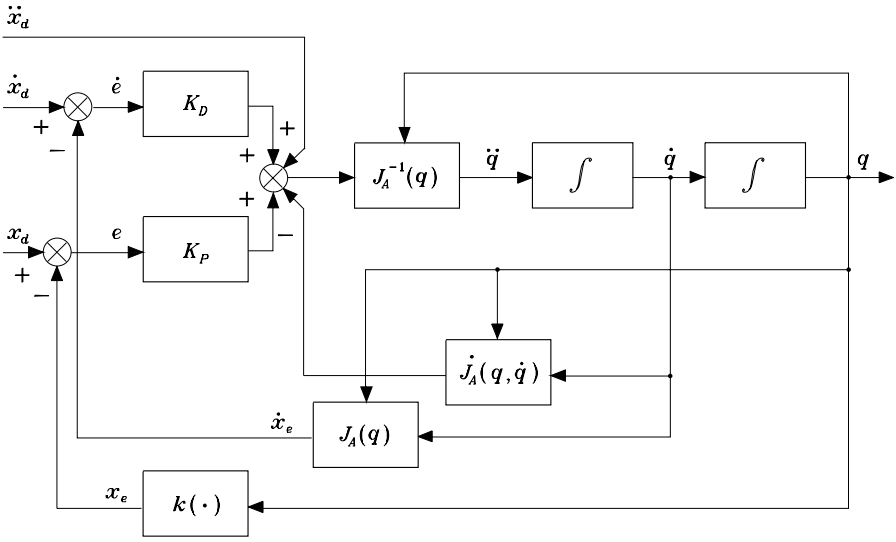

Fig. 3.14. Block scheme of the second-order inverse kinematics algorithm with Jacobian inverse

At this point, it is advisable to choose the joint acceleration vector as

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{K}_P \boldsymbol{e} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} \right) \tag{3.102}$$

where $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ are positive definite (typically diagonal) matrices. Substituting (3.102) into (3.101) leads to the equivalent linear error system

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{K}_P \boldsymbol{e} = \boldsymbol{0} \tag{3.103}$$

which is *asymptotically stable*: the error tends to zero along the trajectory with a convergence speed depending on the choice of the matrices $\boldsymbol{K}_P$ e $\boldsymbol{K}_D$. The second-order inverse kinematics algorithm is illustrated in the block scheme of Fig. 3.14.

In the case of a *redundant manipulator*, the generalization of (3.102) leads to an algorithmic solution based on the Jacobian pseudo-inverse of the kind

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_A^{\dagger} \left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{K}_P \boldsymbol{e} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} \right) + (\boldsymbol{I}_n - \boldsymbol{J}_A^{\dagger} \boldsymbol{J}_A) \ddot{\boldsymbol{q}}_0 \tag{3.104}$$

where the vector $\ddot{\boldsymbol{q}}_0$ represents arbitrary joint accelerations which can be chosen so as to (locally) optimize an objective function like those considered in Sect. 3.5.1.

As for the first-order inverse kinematics algorithms, it is possible to consider other expressions for the orientation error which, unlike the Euler angles, refer to an angle and axis description, else to the unit quaternion.

3.7.5 Comparison Among Inverse Kinematics Algorithms

In order to make a comparison of performance among the inverse kinematics algorithms presented above, consider the 3-link planar arm in Fig. 2.20 whose link lengths are $a_1 = a_2 = a_3 = 0.5\,\text{m}$. The direct kinematics for this arm is given by (2.83), while its Jacobian can be found from (3.35) by considering the 3 non-null rows of interest for the operational space.

Let the arm be at the initial posture $\boldsymbol{q} = [\,\pi \quad -\pi/2 \quad -\pi/2\,]^T\,\text{rad}$, corresponding to the end-effector pose: $\boldsymbol{p} = [\,0 \quad 0.5\,]^T\,\text{m}$, $\phi = 0\,\text{rad}$. A circular path of radius 0.25 m and centre at $(0.25, 0.5)\,\text{m}$ is assigned to the end-effector. Let the motion trajectory be

$$\boldsymbol{p}_d(t) = \begin{bmatrix} 0.25(1 - \cos \pi t) \\ 0.25(2 + \sin \pi t) \end{bmatrix} \qquad 0 \leq t \leq 4;$$

i.e., the end-effector has to make two complete circles in a time of 2 s per circle. As regards end-effector orientation, initially it is required to follow the trajectory

$$\phi_d(t) = \sin \frac{\pi}{24} t \qquad 0 \leq t \leq 4;$$

i.e., the end-effector has to attain a different orientation ($\phi_d = 0.5\,\text{rad}$) at the end of the two circles.

The inverse kinematics algorithms were implemented on a computer by adopting the Euler numerical integration scheme (3.48) with an integration time $\Delta t = 1\,\text{ms}$.

At first, the inverse kinematics along the given trajectory has been performed by using (3.47). The results obtained in Fig. 3.15 show that the norm of the position error along the whole trajectory is bounded; at steady state, after $t = 4$, the error sets to a constant value in view of the typical *drift* of *open-loop* schemes. A similar drift can be observed for the orientation error.

Next, the inverse kinematics algorithm based on (3.70) using the Jacobian *inverse* has been used, with the matrix gain $\boldsymbol{K} = \text{diag}\{500, 500, 100\}$. The resulting joint positions and velocities as well as the tracking errors are shown in Fig. 3.16. The norm of the position error is radically decreased and converges to zero at steady state, thanks to the *closed-loop* feature of the scheme; the orientation error, too, is decreased and tends to zero at steady state.

On the other hand, if the end-effector orientation is not constrained, the operational space becomes two-dimensional and is characterized by the first two rows of the direct kinematics in (2.83) as well as by the Jacobian in (3.36); a *redundant* DOF is then available. Hence, the inverse kinematics algorithm based on (3.72) using the Jacobian *pseudo-inverse* has been used with $\boldsymbol{K} = \text{diag}\{500, 500\}$. If redundancy is not exploited ($\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$), the results in Fig. 3.17 reveal that position tracking remains satisfactory and, of course, the end-effector orientation freely varies along the given trajectory.

With reference to the previous situation, the use of the Jacobian *transpose* algorithm based on (3.76) with $\boldsymbol{K} = \text{diag}\{500, 500\}$ gives rise to a tracking

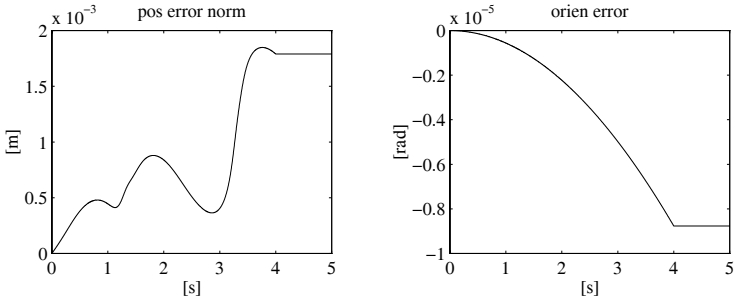

Fig. 3.15. Time history of the norm of end-effector position error and orientation error with the open-loop inverse Jacobian algorithm

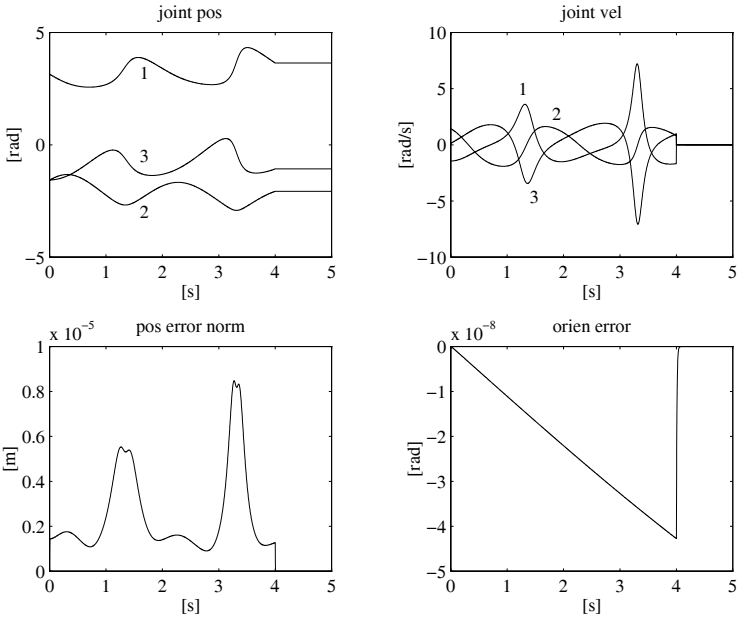

Fig. 3.16. Time history of the joint positions and velocities, and of the norm of end-effector position error and orientation error with the closed-loop inverse Jacobian algorithm

error (Fig. 3.18) which is anyhow bounded and rapidly tends to zero at steady state.

In order to show the capability of handling the degree of redundancy, the algorithm based on (3.72) with $\dot{\boldsymbol{q}}_0 \neq \mathbf{0}$ has been used; two types of constraints

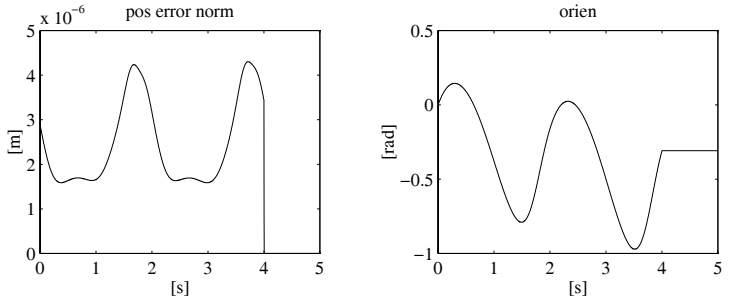

Fig. 3.17. Time history of the norm of end-effector position error and orientation with the Jacobian pseudo-inverse algorithm

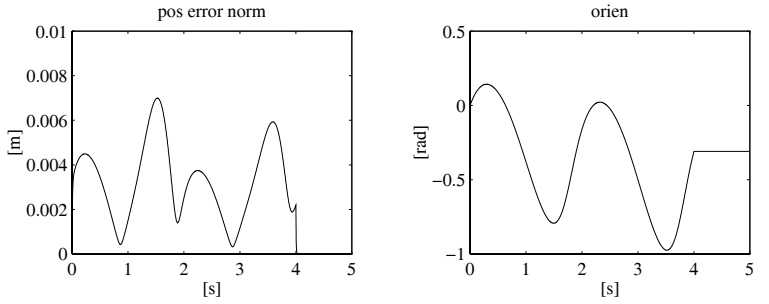

Fig. 3.18. Time history of the norm of end-effector position error and orientation with the Jacobian transpose algorithm

have been considered concerning an objective function to locally maximize according to the choice (3.55). The first function is

$$w(\vartheta_2, \vartheta_3) = \frac{1}{2}(s_2^2 + s_3^2)$$

that provides a *manipulability measure*. Notice that such a function is computationally simpler than the function in (3.56), but it still describes a distance from kinematic singularities in an effective way. The gain in (3.55)) has been set to $k_0 = 50$. In Fig. 3.19, the joint trajectories are reported for the two cases with and without ($k_0 = 0$) constraint. The addition of the constraint leads to having coincident trajectories for Joints 2 and 3. The manipulability measure in the constrained case (*continuous line*) attains larger values along the trajectory compared to the unconstrained case (*dashed line*). It is worth underlining that the tracking position error is practically the same in the two cases (Fig. 3.17), since the additional joint velocity contribution is projected in the null space of the Jacobian so as not to alter the performance of the end-effector position task.

Finally, it is worth noticing that in the constrained case the resulting joint trajectories are *cyclic*, i.e., they take on the same values after a period of

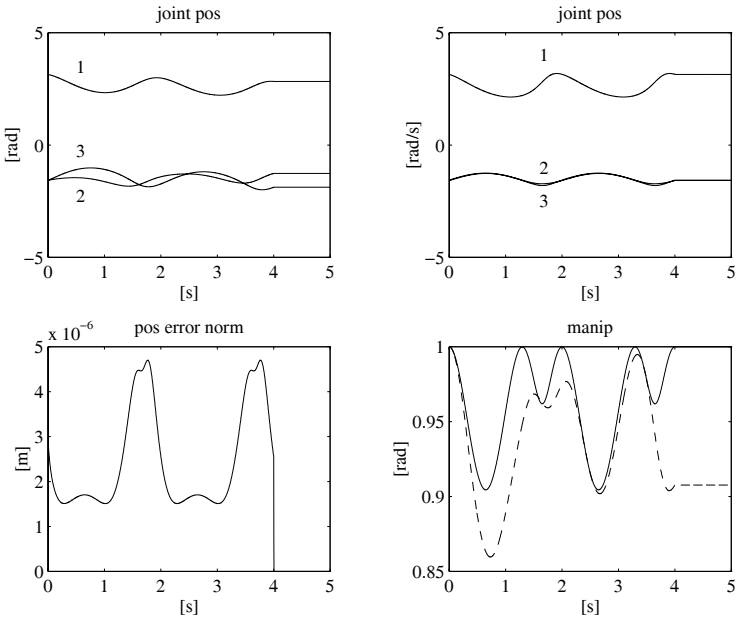

Fig. 3.19. Time history of the joint positions, the norm of end-effector position error, and the manipulability measure with the Jacobian pseudo-inverse algorithm and manipulability constraint; *upper left*: with the unconstrained solution, *upper right*: with the constrained solution

the circular path. This does not happen for the unconstrained case, since the internal motion of the structure causes the arm to be in a different posture after one circle.

The second objective function considered is the *distance from mechanical joint limits* in (3.57). Specifically, it is assumed what follows: the first joint does not have limits ($q_{1m} = -2\pi$, $q_{1M} = 2\pi$), the second joint has limits $q_{2m} = -\pi/2$, $q_{2M} = \pi/2$, and the third joint has limits $q_{3m} = -3\pi/2$, $q_{3M} = -\pi/2$. It is not difficult to verify that, in the unconstrained case, the trajectories of Joints 2 and 3 in Fig. 3.19 violate the respective limits. The gain in (3.55) has been set to $k_0 = 250$. The results in Fig. 3.20 show the effectiveness of the technique with utilization of redundancy, since both Joints 2 and 3 tend to invert their motion — with respect to the unconstrained trajectories in Fig. 3.19 — and keep far from the minimum limit for Joint 2 and the maximum limit for Joint 3, respectively. Such an effort does not appreciably affect the position tracking error, whose norm is bounded anyhow within acceptable values.

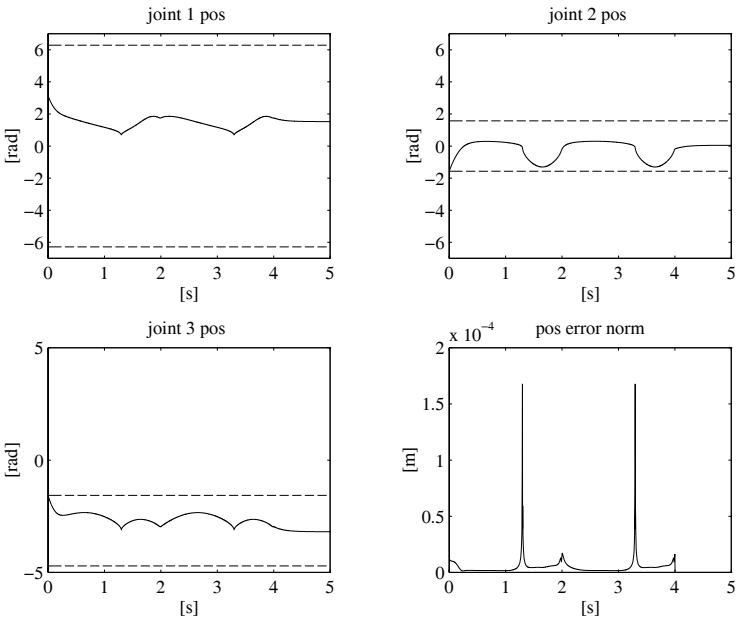

Fig. 3.20. Time history of the joint positions and the norm of end-effector position error with the Jacobian pseudo-inverse algorithm and joint limit constraint (joint limits are denoted by *dashed lines*)

3.8 Statics

The goal of *statics* is to determine the relationship between the generalized forces applied to the end-effector and the generalized forces applied to the joints — forces for prismatic joints, torques for revolute joints — with the manipulator at an equilibrium configuration.

Let $\boldsymbol{\tau}$ denote the $(n \times 1)$ vector of joint torques and $\boldsymbol{\gamma}$ the $(r \times 1)$ vector of end-effector forces[15] where r is the dimension of the operational space of interest.

The application of the *principle of virtual work* allows the determination of the required relationship. The mechanical manipulators considered are systems with time-invariant, holonomic constraints, and thus their configurations depend only on the joint variables $\boldsymbol{q}$ and not explicitly on time. This implies that virtual displacements coincide with elementary displacements.

Consider the elementary works performed by the two force systems. As for the joint torques, the elementary work associated with them is

$$dW_\tau = \boldsymbol{\tau}^T d\boldsymbol{q}. \tag{3.105}$$

[15] Hereafter, generalized forces at the joints are often called *torques*, while generalized forces at the end-effector are often called *forces*.

As for the end-effector forces $\boldsymbol{\gamma}$, if the force contributions $\boldsymbol{f}_e$ are separated by the moment contributions $\boldsymbol{\mu}_e$, the elementary work associated with them is

$$dW_\gamma = \boldsymbol{f}_e^T d\boldsymbol{p}_e + \boldsymbol{\mu}_e^T \boldsymbol{\omega}_e dt, \tag{3.106}$$

where $d\boldsymbol{p}_e$ is the linear displacement and $\boldsymbol{\omega}_e dt$ is the angular displacement[16]

By accounting for the differential kinematics relationship in (3.4), (3.5), the relation (3.106) can be rewritten as

$$\begin{aligned} dW_\gamma &= \boldsymbol{f}_e^T \boldsymbol{J}_P(\boldsymbol{q}) d\boldsymbol{q} + \boldsymbol{\mu}_e^T \boldsymbol{J}_O(\boldsymbol{q}) d\boldsymbol{q} \\ &= \boldsymbol{\gamma}_e^T \boldsymbol{J}(\boldsymbol{q}) d\boldsymbol{q} \end{aligned} \tag{3.107}$$

where $\boldsymbol{\gamma}_e = [\, \boldsymbol{f}_e^T \quad \boldsymbol{\mu}_e^T \,]^T$. Since virtual and elementary displacements coincide, the virtual works associated with the two force systems are

$$\delta W_\tau = \boldsymbol{\tau}^T \delta \boldsymbol{q} \tag{3.108}$$

$$\delta W_\gamma = \boldsymbol{\gamma}_e^T \boldsymbol{J}(\boldsymbol{q}) \delta \boldsymbol{q}, \tag{3.109}$$

where δ is the usual symbol to indicate virtual quantities.

According to the principle of virtual work, the manipulator is at *static equilibrium* if and only if

$$\delta W_\tau = \delta W_\gamma \qquad \forall \delta \boldsymbol{q}, \tag{3.110}$$

i.e., the difference between the virtual work of the joint torques and the virtual work of the end-effector forces must be null for all joint displacements.

From (3.109), notice that the virtual work of the end-effector forces is null for any displacement in the null space of $\boldsymbol{J}$. This implies that the joint torques associated with such displacements must be null at static equilibrium. Substituting (3.108), (3.109) into (3.110) leads to the notable result

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q}) \boldsymbol{\gamma}_e \tag{3.111}$$

stating that the relationship between the end-effector forces and the joint torques is established by the transpose of the manipulator geometric Jacobian.

3.8.1 Kineto-Statics Duality

The statics relationship in (3.111), combined with the differential kinematics equation in (3.45), points out a property of *kineto-statics duality*. In fact, by adopting a representation similar to that of Fig. 3.7 for differential kinematics, one has that (Fig. 3.21):

- The range space of $\boldsymbol{J}^T$ is the subspace $\mathcal{R}(\boldsymbol{J}^T)$ in $\mathbb{R}^n$ of the joint torques that can balance the end-effector forces, in the given manipulator posture.

[16] The angular displacement has been indicated by $\boldsymbol{\omega}_e dt$ in view of the problems of integrability of $\boldsymbol{\omega}_e$ discussed in Sect. 3.6.

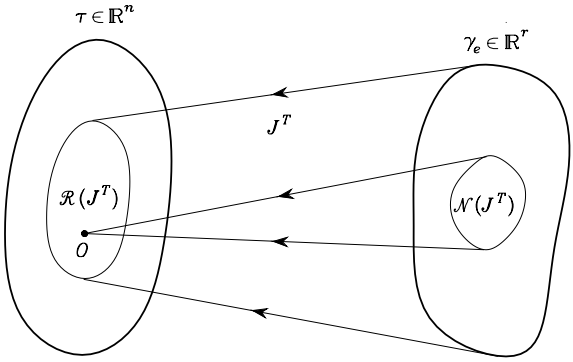

Fig. 3.21. Mapping between the end-effector force space and the joint torque space

- The null space of $\boldsymbol{J}^T$ is the subspace $\mathcal{N}(\boldsymbol{J}^T)$ in $\mathbb{R}^r$ of the end-effector forces that do not require any balancing joint torques, in the given manipulator posture.

It is worth remarking that the end-effector forces $\boldsymbol{\gamma}_e \in \mathcal{N}(\boldsymbol{J}^T)$ are entirely absorbed by the structure in that the mechanical constraint reaction forces can balance them exactly. Hence, a manipulator at a singular configuration remains in the given posture whatever end-effector force $\boldsymbol{\gamma}_e$ is applied so that $\boldsymbol{\gamma}_e \in \mathcal{N}(\boldsymbol{J}^T)$.

The relations between the two subspaces are established by

$$\mathcal{N}(\boldsymbol{J}) \equiv \mathcal{R}^{\perp}(\boldsymbol{J}^T) \qquad \mathcal{R}(\boldsymbol{J}) \equiv \mathcal{N}^{\perp}(\boldsymbol{J}^T)$$

and then, once the manipulator Jacobian is known, it is possible to characterize completely differential kinematics and statics in terms of the range and null spaces of the Jacobian and its transpose.

On the basis of the above duality, the inverse kinematics scheme with the Jacobian transpose in Fig. 3.12 admits an interesting physical interpretation. Consider a manipulator with ideal dynamics $\boldsymbol{\tau} = \dot{\boldsymbol{q}}$ (null masses and unit viscous friction coefficients); the algorithm update law $\dot{\boldsymbol{q}} = \boldsymbol{J}^T \boldsymbol{K} \boldsymbol{e}$ plays the role of a generalized spring of stiffness constant $\boldsymbol{K}$ generating a force $\boldsymbol{K}\boldsymbol{e}$ that pulls the end-effector towards the desired posture in the operational space. If this manipulator is allowed to move, e.g., in the case $\boldsymbol{K}\boldsymbol{e} \notin \mathcal{N}(\boldsymbol{J}^T)$, the end-effector attains the desired posture and the corresponding joint variables are determined.

3.8.2 Velocity and Force Transformation

The kineto-statics duality concept presented above can be useful to characterize the transformation of velocities and forces between two coordinate frames.

Consider a reference coordinate frame O_0–$x_0y_0z_0$ and a rigid body moving with respect to such a frame. Then let O_1–$x_1y_1z_1$ and O_2–$x_2y_2z_2$ be two

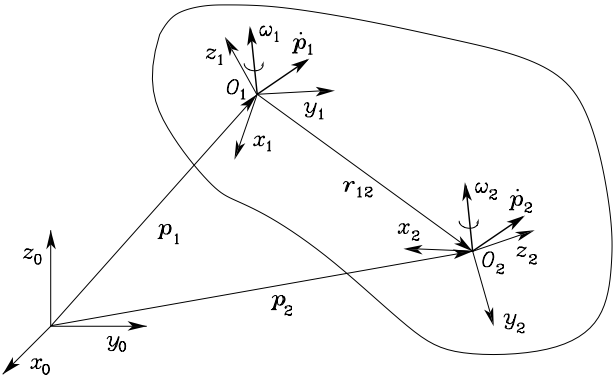

Fig. 3.22. Representation of linear and angular velocities in different coordinate frames on the same rigid body

coordinate frames attached to the body (Fig. 3.22). The relationships between translational and rotational velocities of the two frames with respect to the reference frame are given by

$$\begin{aligned}\boldsymbol{\omega}_2 &= \boldsymbol{\omega}_1\\ \dot{\boldsymbol{p}}_2 &= \dot{\boldsymbol{p}}_1 + \boldsymbol{\omega}_1 \times \boldsymbol{r}_{12}.\end{aligned}$$

By exploiting the skew-symmetric operator $\boldsymbol{S}(\cdot)$ in (3.9), the above relations can be compactly written as

$$\begin{bmatrix}\dot{\boldsymbol{p}}_2\\ \boldsymbol{\omega}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}(\boldsymbol{r}_{12})\\ \boldsymbol{O} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{p}}_1\\ \boldsymbol{\omega}_1\end{bmatrix}. \tag{3.112}$$

All vectors in (3.112) are meant to be referred to the reference frame O_0–$x_0y_0z_0$. On the other hand, if vectors are referred to their own frames, it is

$$\boldsymbol{r}_{12} = \boldsymbol{R}_1\boldsymbol{r}_{12}^1$$

and also

$$\begin{aligned}\dot{\boldsymbol{p}}_1 &= \boldsymbol{R}_1\dot{\boldsymbol{p}}_1^1 \qquad \dot{\boldsymbol{p}}_2 = \boldsymbol{R}_2\dot{\boldsymbol{p}}_2^2 = \boldsymbol{R}_1\boldsymbol{R}_2^1\dot{\boldsymbol{p}}_2^2\\ \boldsymbol{\omega}_1 &= \boldsymbol{R}_1\boldsymbol{\omega}_1^1 \qquad \boldsymbol{\omega}_2 = \boldsymbol{R}_2\boldsymbol{\omega}_2^2 = \boldsymbol{R}_1\boldsymbol{R}_2^1\boldsymbol{\omega}_2^2.\end{aligned}$$

Accounting for (3.112) and (3.11) gives

$$\begin{aligned}\boldsymbol{R}_1\boldsymbol{R}_2^1\dot{\boldsymbol{p}}_2^2 &= \boldsymbol{R}_1\dot{\boldsymbol{p}}_1^1 - \boldsymbol{R}_1\boldsymbol{S}(\boldsymbol{r}_{12}^1)\boldsymbol{R}_1^T\boldsymbol{R}_1\boldsymbol{\omega}_1^1\\ \boldsymbol{R}_1\boldsymbol{R}_2^1\boldsymbol{\omega}_2^2 &= \boldsymbol{R}_1\boldsymbol{\omega}_1^1.\end{aligned}$$

Eliminating the dependence on $\boldsymbol{R}_1$, which is premultiplied to each term on both sides of the previous relations, yields[17]

$$\begin{bmatrix}\dot{\boldsymbol{p}}_2^2\\ \boldsymbol{\omega}_2^2\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}_1^2 & -\boldsymbol{R}_1^2\boldsymbol{S}(\boldsymbol{r}_{12}^1)\\ \boldsymbol{O} & \boldsymbol{R}_1^2\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{p}}_1^1\\ \boldsymbol{\omega}_1^1\end{bmatrix} \tag{3.113}$$

[17] Recall that $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$, as in (2.4).

giving the sought general relationship of *velocity transformation* between two frames.

It may be observed that the transformation matrix in (3.113) plays the role of a true Jacobian, since it characterizes a velocity transformation, and thus (3.113) may be shortly written as

$$\boldsymbol{v}_2^2 = \boldsymbol{J}_1^2 \boldsymbol{v}_1^1. \tag{3.114}$$

At this point, by virtue of the kineto-statics duality, the *force transformation* between two frames can be directly derived in the form

$$\boldsymbol{\gamma}_1^1 = \boldsymbol{J}_1^{2T} \boldsymbol{\gamma}_2^2 \tag{3.115}$$

which can be detailed into[18]

$$\begin{bmatrix} \boldsymbol{f}_1^1 \\ \boldsymbol{\mu}_1^1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_2^1 & \boldsymbol{O} \\ \boldsymbol{S}(\boldsymbol{r}_{12}^1)\boldsymbol{R}_2^1 & \boldsymbol{R}_2^1 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_2^2 \\ \boldsymbol{\mu}_2^2 \end{bmatrix}. \tag{3.116}$$

Finally, notice that the above analysis is instantaneous in that, if a coordinate frame varies with respect to the other, it is necessary to recompute the Jacobian of the transformation through the computation of the related rotation matrix of one frame with respect to the other.

3.8.3 Closed Chain

As discussed in Sect. 2.8.3, whenever the manipulator contains a closed chain, there is a functional relationship between the joint variables. In particular, the closed chain structure is transformed into a tree-structured open chain by virtually cutting the loop at a joint. It is worth choosing such a cut joint as one of the unactuated joints. Then, the constraints (2.59) or (2.60) should be solved for a reduced number of joint variables, corresponding to the DOFs of the chain. Therefore, it is reasonable to assume that at least such independent joints are actuated, while the others may or may not be actuated. Let $\boldsymbol{q}_o = [\, \boldsymbol{q}_a^T \quad \boldsymbol{q}_u^T \,]^T$ denote the vector of joint variables of the tree-structured open chain, where $\boldsymbol{q}_a$ and $\boldsymbol{q}_u$ are the vectors of *actuated* and *unactuated* joint variables, respectively. Assume that from the above constraints it is possible to determine a functional expression

$$\boldsymbol{q}_u = \boldsymbol{q}_u(\boldsymbol{q}_a). \tag{3.117}$$

Time differentiation of (3.117) gives the relationship between joint velocities in the form

$$\dot{\boldsymbol{q}}_o = \boldsymbol{\Upsilon} \dot{\boldsymbol{q}}_a \tag{3.118}$$

where

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{I} \\ \dfrac{\partial \boldsymbol{q}_u}{\partial \boldsymbol{q}_a} \end{bmatrix} \tag{3.119}$$

[18] The skew-symmetry property $\boldsymbol{S} + \boldsymbol{S}^T = \boldsymbol{O}$ is utilized.

is the transformation matrix between the two vectors of joint velocities, which in turn plays the role of a Jacobian.

At this point, according to an intuitive kineto-statics duality concept, it is possible to describe the transformation between the corresponding vectors of joint torques in the form

$$\boldsymbol{\tau}_a = \boldsymbol{\Upsilon}^T \boldsymbol{\tau}_o \tag{3.120}$$

where $\boldsymbol{\tau}_o = [\, \boldsymbol{\tau}_a^T \quad \boldsymbol{\tau}_u^T \,]^T$, with obvious meaning of the quantities.

Example 3.5

Consider the parallelogram arm of Sect. 2.9.2. On the assumption to actuate the two Joints $1'$ and $1''$ at the base, it is $\boldsymbol{q}_a = [\, \vartheta_{1'} \quad \vartheta_{1''} \,]^T$ and $\boldsymbol{q}_u = [\, \vartheta_{2'} \quad \vartheta_{3'} \,]^T$. Then, using (2.64), the transformation matrix in (3.119) is

$$\boldsymbol{\Upsilon} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Hence, in view of (3.120), the torque vector of the actuated joints is

$$\boldsymbol{\tau}_a = \begin{bmatrix} \tau_{1'} - \tau_{2'} + \tau_{3'} \\ \tau_{1''} + \tau_{2'} - \tau_{3'} \end{bmatrix} \tag{3.121}$$

while obviously $\boldsymbol{\tau}_u = [\, 0 \quad 0 \,]^T$ in agreement with the fact that both Joints $2'$ and $3'$ are unactuated.

3.9 Manipulability Ellipsoids

The differential kinematics equation in (3.45) and the statics equation in (3.111), together with the duality property, allow the definition of indices for the evaluation of manipulator performance. Such indices can be helpful both for mechanical manipulator design and for determining suitable manipulator postures to execute a given task in the current configuration.

First, it is desired to represent the attitude of a manipulator to arbitrarily change end-effector position and orientation. This capability is described in an effective manner by the *velocity manipulability ellipsoid*.

Consider the set of joint velocities of constant (unit) norm

$$\dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} = 1; \tag{3.122}$$

this equation describes the points on the surface of a sphere in the joint velocity space. It is desired to describe the operational space velocities that can

be generated by the given set of joint velocities, with the manipulator in a given posture. To this end, one can utilize the differential kinematics equation in (3.45) solved for the joint velocities; in the general case of a redundant manipulator ($r < n$) at a nonsingular configuration, the minimum-norm solution $\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{v}_e$ can be considered which, substituted into (3.122), yields

$$\boldsymbol{v}_e^T \big(\boldsymbol{J}^{\dagger T}(\boldsymbol{q})\boldsymbol{J}^{\dagger}(\boldsymbol{q})\big)\boldsymbol{v}_e = 1.$$

Accounting for the expression of the pseudo-inverse of $\boldsymbol{J}$ in (3.52) gives

$$\boldsymbol{v}_e^T \big(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\big)^{-1}\boldsymbol{v}_e = 1, \tag{3.123}$$

which is the equation of the points on the surface of an ellipsoid in the end-effector velocity space.

The choice of the minimum-norm solution rules out the presence of internal motions for the redundant structure. If the general solution (3.54) is used for $\dot{\boldsymbol{q}}$, the points satisfying (3.122) are mapped into points inside the ellipsoid whose surface is described by (3.123).

For a nonredundant manipulator, the differential kinematics solution (3.47) is used to derive (3.123); in this case the points on the surface of the sphere in the joint velocity space are mapped into points on the surface of the ellipsoid in the end-effector velocity space.

Along the direction of the major axis of the ellipsoid, the end-effector can move at large velocity, while along the direction of the minor axis small end-effector velocities are obtained. Further, the closer the ellipsoid is to a sphere — unit eccentricity — the better the end-effector can move isotropically along all directions of the operational space. Hence, it can be understood why this ellipsoid is an index characterizing manipulation ability of the structure in terms of velocities.

As can be recognized from (3.123), the shape and orientation of the ellipsoid are determined by the core of its quadratic form and then by the matrix $\boldsymbol{J}\boldsymbol{J}^T$ which is in general a function of the manipulator configuration. The directions of the principal axes of the ellipsoid are determined by the eigenvectors $\boldsymbol{u}_i$, for $i = 1, \ldots, r$, of the matrix $\boldsymbol{J}\boldsymbol{J}^T$, while the dimensions of the axes are given by the singular values of $\boldsymbol{J}$, $\sigma_i = \sqrt{\lambda_i(\boldsymbol{J}\boldsymbol{J}^T)}$, for $i = 1, \ldots, r$, where $\lambda_i(\boldsymbol{J}\boldsymbol{J}^T)$ denotes the generic eigenvalue of $\boldsymbol{J}\boldsymbol{J}^T$.

A global representative measure of manipulation ability can be obtained by considering the volume of the ellipsoid. This volume is proportional to the quantity

$$w(\boldsymbol{q}) = \sqrt{\det\big(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\big)}$$

which is the *manipulability measure* already introduced in (3.56). In the case of a nonredundant manipulator ($r = n$), w reduces to

$$w(\boldsymbol{q}) = \big|\det\big(\boldsymbol{J}(\boldsymbol{q})\big)\big|. \tag{3.124}$$

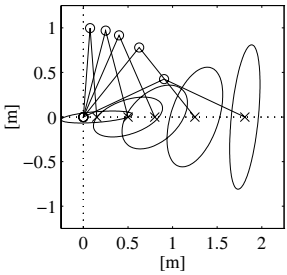

Fig. 3.23. Velocity manipulability ellipses for a two-link planar arm in different postures

It is easy to recognize that it is always $w > 0$, except for a manipulator at a singular configuration when $w = 0$. For this reason, this measure is usually adopted as a distance of the manipulator from singular configurations.

Example 3.6

Consider the two-link planar arm. From the expression in (3.41), the manipulability measure is in this case

$$w = |\det(\boldsymbol{J})| = a_1 a_2 |s_2|.$$

Therefore, as a function of the arm postures, the manipulability is maximum for $\vartheta_2 = \pm\pi/2$. On the other hand, for a given constant reach $a_1 + a_2$, the structure offering the maximum manipulability, independently of ϑ_1 and ϑ_2, is the one with $a_1 = a_2$.

These results have a biomimetic interpretation in the human arm, if that is regarded as a two-link arm (arm + forearm). The condition $a_1 = a_2$ is satisfied with good approximation. Further, the elbow angle ϑ_2 is usually in the neighbourhood of $\pi/2$ in the execution of several tasks, such as that of writing. Hence, the human being tends to dispose the arm in the most dexterous configuration from a manipulability viewpoint.

Figure 3.23 illustrates the velocity manipulability ellipses for a certain number of postures with the tip along the horizontal axis and $a_1 = a_2 = 1$. It can be seen that when the arm is outstretched the ellipsoid is very thin along the vertical direction. Hence, one recovers the result anticipated in the study of singularities that the arm in this posture can generate tip velocities preferably along the vertical direction. In Fig. 3.24, moreover, the behaviour of the minimum and maximum singular values of the matrix $\boldsymbol{J}$ is illustrated as a function of tip position along axis x; it can be verified that the minimum singular value is null when the manipulator is at a singularity (retracted or outstretched).

Therefore, with reference to the postures, manipulability has a maximum for $\vartheta_2 = \pm\pi/2$. On the other hand, for a given total extension $a_1 + a_2$, the structure which, independently of ϑ_1 and ϑ_2, offers the largest manipulability is that with $a_1 = a_2$.

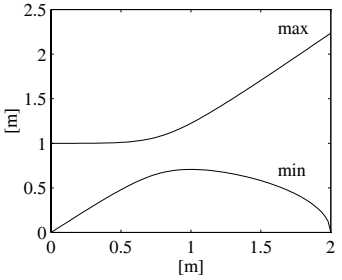

Fig. 3.24. Minimum and maximum singular values of $\boldsymbol{J}$ for a two-link planar arm as a function of the arm posture

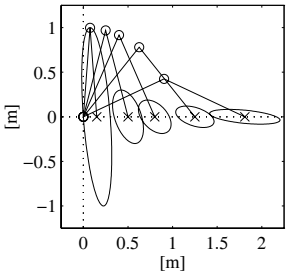

Fig. 3.25. Force manipulability ellipses for a two-link planar arm in different postures

The manipulability measure w has the advantage of being easy to compute, through the determinant of matrix $\boldsymbol{J}\boldsymbol{J}^T$. However, its numerical value does not constitute an absolute measure of the actual closeness of the manipulator to a singularity. It is enough to consider the above example and take two arms of identical structure, one with links of 1 m and the other with links of 1 cm. Two different values of manipulability are obtained which differ by four orders of magnitude. Hence, in that case it is convenient to consider only $|s_2|$ — eventually $|\vartheta_2|$ — as the manipulability measure. In more general cases when it is not easy to find a simple, meaningful index, one can consider the ratio between the minimum and maximum singular values of the Jacobian σ_r/σ_1 which is equivalent to the inverse of the condition number of matrix $\boldsymbol{J}$. This ratio gives not only a measure of the distance from a singularity ($\sigma_r = 0$), but also a direct measure of eccentricity of the ellipsoid. The disadvantage in utilizing this index is its computational complexity; it is practically impossible to compute it in symbolic form, i.e., as a function of the joint configuration, except for matrices of reduced dimension.

On the basis of the existing duality between differential kinematics and statics, it is possible to describe the manipulability of a structure not only

with reference to velocities, but also with reference to forces. To be specific, one can consider the sphere in the space of joint torques

$$\boldsymbol{\tau}^T\boldsymbol{\tau} = 1 \tag{3.125}$$

which, accounting for (3.111), is mapped into the ellipsoid in the space of end-effector forces

$$\boldsymbol{\gamma}_e^T\big(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\big)\boldsymbol{\gamma}_e = 1 \tag{3.126}$$

which is defined as the *force manipulability ellipsoid*. This ellipsoid characterizes the end-effector forces that can be generated with the given set of joint torques, with the manipulator in a given posture.

As can be easily recognized from (3.126), the core of the quadratic form is constituted by the inverse of the matrix core of the velocity ellipsoid in (3.123). This feature leads to the notable result that the principal axes of the force manipulability ellipsoid coincide with the principal axes of the velocity manipulability ellipsoid, while the dimensions of the respective axes are in inverse proportion. Therefore, according to the concept of force/velocity duality, a direction along which good velocity manipulability is obtained is a direction along which poor force manipulability is obtained, and vice versa.

In Fig. 3.25, the manipulability ellipses for the same postures as those of the example in Fig. 3.23 are illustrated. A comparison of the shape and orientation of the ellipses confirms the force/velocity duality effect on the manipulability along different directions.

It is worth pointing out that these manipulability ellipsoids can be represented geometrically in all cases of an operational space of dimension at most 3. Therefore, if it is desired to analyze manipulability in a space of greater dimension, it is worth separating the components of linear velocity (force) from those of angular velocity (moment), also avoiding problems due to non-homogeneous dimensions of the relevant quantities (e.g., m/s vs rad/s). For instance, for a manipulator with a spherical wrist, the manipulability analysis is naturally prone to a decoupling between arm and wrist.

An effective interpretation of the above results can be achieved by regarding the manipulator as a *mechanical transformer* of velocities and forces from the joint space to the operational space. Conservation of energy dictates that an amplification in the velocity transformation is necessarily accompanied by a reduction in the force transformation, and vice versa. The transformation ratio along a given direction is determined by the intersection of the vector along that direction with the surface of the ellipsoid. Once a unit vector $\boldsymbol{u}$ along a direction has been assigned, it is possible to compute the transformation ratio for the force manipulability ellipsoid as

$$\alpha(\boldsymbol{q}) = \left(\boldsymbol{u}^T\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{u}\right)^{-1/2} \tag{3.127}$$

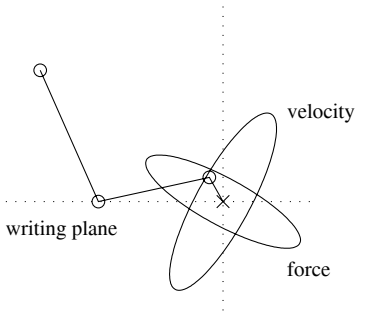

Fig. 3.26. Velocity and force manipulability ellipses for a 3-link planar arm in a typical configuration for a task of controlling force and velocity

and for the velocity manipulability ellipsoid as

$$\beta(\boldsymbol{q}) = \left(\boldsymbol{u}^T \left(\boldsymbol{J}(\boldsymbol{q}) \boldsymbol{J}^T(\boldsymbol{q}) \right)^{-1} \boldsymbol{u} \right)^{-1/2}. \tag{3.128}$$

The manipulability ellipsoids can be conveniently utilized not only for analyzing manipulability of the structure along different directions of the operational space, but also for determining compatibility of the structure to execute a task assigned along a direction. To this end, it is useful to distinguish between actuation tasks and control tasks of velocity and force. In terms of the relative ellipsoid, the task of actuating a velocity (force) requires preferably a large transformation ratio along the task direction, since for a given set of joint velocities (forces) at the joints it is possible to generate a large velocity (force) at the end-effector. On the other hand, for a control task it is important to have a small transformation ratio so as to gain good sensitivity to errors that may occur along the given direction.

Revisiting once again the duality between velocity manipulability ellipsoid and force manipulability ellipsoid, it can be found that an optimal direction to actuate a velocity is also an optimal direction to control a force. Analogously, a good direction to actuate a force is also a good direction to control a velocity.

To have a tangible example of the above concept, consider the typical task of writing on a horizontal surface for the human arm; this time, the arm is regarded as a 3-link planar arm: arm + forearm + hand. Restricting the analysis to a two-dimensional task space (the direction vertical to the surface and the direction of the line of writing), one has to achieve fine control of the vertical force (the pressure of the pen on the paper) and of the horizontal velocity (to write in good calligraphy). As a consequence, the force manipulability ellipse tends to be oriented horizontally for correct task execution. Correspondingly, the velocity manipulability ellipse tends to be oriented vertically in perfect agreement with the task requirement. In this case, from Fig. 3.26 the typical configuration of the human arm when writing can be recognized.

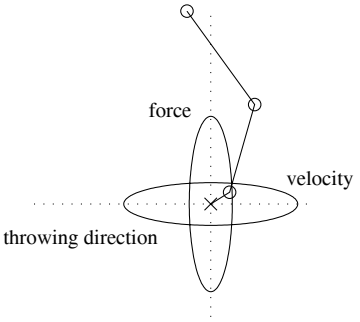

Fig. 3.27. Velocity and force manipulability ellipses for a 3-link planar arm in a typical configuration for a task of actuating force and velocity

An opposite example to the previous one is that of the human arm when throwing a weight in the horizontal direction. In fact, now it is necessary to actuate a large vertical force (to sustain the weight) and a large horizontal velocity (to throw the load for a considerable distance). Unlike the above, the force (velocity) manipulability ellipse tends to be oriented vertically (horizontally) to successfully execute the task. The relative configuration in Fig. 3.27 is representative of the typical attitude of the human arm when, for instance, releasing the ball in a bowling game.

In the above two examples, it is worth pointing out that the presence of a two-dimensional operational space is certainly advantageous to try reconfiguring the structure in the best configuration compatible with the given task. In fact, the transformation ratios defined in (3.127) and (3.128) are scalar functions of the manipulator configurations that can be optimized locally according to the technique for exploiting redundant DOFs previously illustrated.

Bibliography

The concept of geometric Jacobian was originally introduced in [240] and the problem of its computationally efficient determination is considered in [173]. The concept of analytical Jacobian is presented in [114] with reference to operational space control.

Inverse differential kinematics dates back to [240] under the name of resolved rate control. The use of the Jacobian pseudo-inverse is due to [118]. The adoption of the damped least-squares inverse has been independently proposed by [161] and [238]; a tutorial on the topic is [42]. The inverse kinematics algorithm based on the Jacobian transpose has been originally proposed in [198, 16]. Further details about the orientation error are found in [142, 250, 132, 41].

The utilization of the joint velocities in the null space of the Jacobian for redundancy resolution is proposed in [129] and further refined in [147] regarding the choice of the objective functions. The approach based on task priority

is presented in [163]; other approaches based on the concept of augmented task space are presented in [14, 69, 199, 203, 194, 37]. For global redundancy resolutions see [162]. A complete treatment of redundant manipulators can be found in [160] while a tutorial is [206].

The extension of inverse kinematics to the second order has been proposed in [207], while the symbolic differentiation of the solutions in terms of joint velocities to obtain stable acceleration solutions can be found in [208]. Further details about redundancy resolution are in [59].

The concepts of kineto-statics duality are discussed in [191]. The manipulability ellipsoids are proposed in [245, 248] and employed in [44] for posture dexterity analysis with regard to manipulation tasks.

Problems

3.1. Prove (3.11).

3.2. Compute the Jacobian of the cylindrical arm in Fig. 2.35.

3.3. Compute the Jacobian of the SCARA manipulator in Fig. 2.36.

3.4. Find the singularities of the 3-link planar arm in Fig. 2.20.

3.5. Find the singularities of the spherical arm in Fig. 2.22.

3.6. Find the singularities of the cylindrical arm in Fig. 2.35.

3.7. Find the singularities of the SCARA manipulator in Fig. 2.36.

3.8. Show that the manipulability measure defined in (3.56) is given by the product of the singular values of the Jacobian matrix.

3.9. For the 3-link planar arm in Fig. 2.20, find an expression of the distance of the arm from a circular obstacle of given radius and coordinates.

3.10. Find the solution to the differential kinematics equation with the damped least-square inverse in (3.59).

3.11. Prove (3.64) in an alternative way, i.e., by computing $\boldsymbol{S}(\boldsymbol{\omega}_e)$ as in (3.6) starting from $\boldsymbol{R}(\boldsymbol{\phi})$ in (2.18).

3.12. With reference to (3.64), find the transformation matrix $\boldsymbol{T}(\boldsymbol{\phi}_e)$ in the case of RPY angles.

3.13. With reference to (3.64), find the triplet of Euler angles for which $\boldsymbol{T}(\boldsymbol{0}) = \boldsymbol{I}$.

3.14. Show how the inverse kinematics scheme of Fig. 3.11 can be simplified in the case of a manipulator having a spherical wrist.

3.15. Find an expression of the upper bound on the norm of $\boldsymbol{e}$ for the solution (3.76) in the case $\dot{\boldsymbol{x}}_d \neq \boldsymbol{0}$.

3.16. Prove (3.85).

3.17. Prove (3.86), (3.87).

3.18. Prove that the equation relating the angular velocity to the time derivative of the quaternion is given by

$$\boldsymbol{\omega} = 2\boldsymbol{S}(\boldsymbol{\epsilon})\dot{\boldsymbol{\epsilon}} + 2\eta\dot{\boldsymbol{\epsilon}} - 2\dot{\eta}\boldsymbol{\epsilon}.$$

[*Hint*: Start by showing that (2.33) can be rewritten as $\boldsymbol{R}(\eta, \boldsymbol{\epsilon}) = (2\eta^2 - 1)\boldsymbol{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + 2\eta\boldsymbol{S}(\boldsymbol{\epsilon})$].

3.19. Prove (3.94), (3.95).

3.20. Prove that the time derivative of the Lyapunov function in (3.96) is given by (3.97).

3.21. Consider the 3-link planar arm in Fig. 2.20, whose link lengths are respectively 0.5 m, 0.3 m, 0.3 m. Perform a computer implementation of the inverse kinematics algorithm using the Jacobian pseudo-inverse along the operational space path given by a straight line connecting the points of coordinates $(0.8, 0.2)$ m and $(0.8, -0.2)$ m. Add a constraint aimed at avoiding link collision with a circular object located at $\emptyset = [\,0.3 \quad 0\,]^T$ m of radius 0.1 m. The initial arm configuration is chosen so that $\boldsymbol{p}_e(0) = \boldsymbol{p}_d(0)$. The final time is 2 s. Use sinusoidal motion timing laws. Adopt the Euler numerical integration scheme (3.48) with an integration time $\Delta t = 1$ ms.

3.22. Consider the SCARA manipulator in Fig. 2.36, whose links both have a length of 0.5 m and are located at a height of 1 m from the supporting plane. Perform a computer implementation of the inverse kinematics algorithms with both Jacobian inverse and Jacobian transpose along the operational space path whose position is given by a straight line connecting the points of coordinates $(0.7, 0, 0)$ m and $(0, 0.8, 0.5)$ m, and whose orientation is given by a rotation from 0 rad to $\pi/2$ rad. The initial arm configuration is chosen so that $\boldsymbol{x}_e(0) = \boldsymbol{x}_d(0)$. The final time is 2 s. Use sinusoidal motion timing laws. Adopt the Euler numerical integration scheme (3.48) with an integration time $\Delta t = 1$ ms.

3.23. Prove that the directions of the principal axes of the force and velocity manipulability ellipsoids coincide while their dimensions are in inverse proportion.