CHAPTER

2

Kinematics

2.1 INTRODUCTION

This chapter discusses the kinematics of humanoid robots. Kinematics plays a basic role in motion analysis, generation, and control. Kinematics-based motion control is quite often employed in control algorithms for hand/feet movements.

The chapter is divided in eleven sections. In Section 2.2, the kinematic structure of a generic humanoid robot is presented and main coordinate frames of interest are defined. In Section 2.3, the forward and inverse kinematics problems are addressed via notions for rigid-body position/orientation. Section 2.4 focuses on the forward and inverse differential kinematic relations, including spatial velocity and spatial transformations. The special case of differential kinematic relations at singular configurations is discussed in Section 2.5. Generic singular configurations for robots with and without kinematically redundant limbs are highlighted. Section 2.6 introduces the manipulability ellipsoid. Further on, a number of humanoids comprise kinematically redundant arms and/or legs. Kinematic redundancy resolution is discussed in Section 2.7, focusing on the limbs. Section 2.8 analyzes the important problem of redundancy resolution under motion constraints imposed by multiple motion tasks. This problem is related to whole-body motion control problem and is tackled in Chapter 4 and Chapter 5. Section 2.9 introduces motion constraints due to physical contacts between the robot links and the environment. Differential motion relationships within closed kinematic loops resulting from the contacts are discussed in Section 2.10. Finally, the differential motion relationships of a humanoid robot are derived in Section 2.11.

It is assumed that readers are familiar with the basic concepts concerning systems of rigid bodies, such as representations of positions and orientations, joint models, coordinate frame assignment techniques for serial-link and parallel-link mechanisms, coordinate transformations, and the forward and inverse kinematics problems.

2.2 KINEMATIC STRUCTURE

The skeleton figure of a generic humanoid robot is shown in Fig. 2.1A. The anthropomorphic structure of the robot becomes apparent from the figure: there are six-degree-of-freedom (6-DoF) legs, 7-DoF arms, a 2-DoF torso, and a 2-DoF head. In the real robot, all joints are single-DoF rotational R joints. Robot models, on the other hand, quite often make use of

https://doi.org/10.1016/B978-0-12-804560-2.00009-2

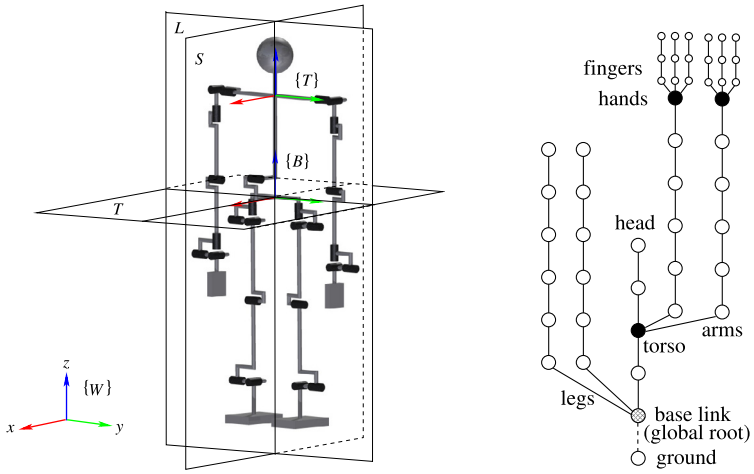

FIGURE 2.1 (A) Left: Skeleton figure of a humanoid robot with three basic reference planes derived from human anatomy and three reference frames. The coloring of the coordinate axes applies henceforth throughout the text. (B) Right: Tree connectivity structure of the kinematic chain. The black circles denote local root links for the branches. The white circles denote the joints.

equivalent multi-DoF joints. In this case, in resemblance to the human skeleton structure, the 3R joint assemblies at the hips, shoulders, and wrists are represented as equivalent spherical joints (3-DoF). The 2R joint assemblies at the ankles, torso, and neck, on the other hand, constitute equivalent universal joints (2-DoF). All robot links in the skeleton figure are assumed to represent rigid bodies. The position/orientation of each link can then be assessed via the position/orientation of a coordinate frame embedded into the link in an appropriate way. For example, with single-DoF models, the Denavit and Hartenberg notation [26] is frequently used. With multi-DoF joint models, on the other hand, the link coordinate frame is attached to one of the connecting joints. Details will be given at a later point.

One of the links in the kinematic chain, the "pelvis" link, plays a special role. Note that the kinematic chain of a humanoid robot can be expressed in the form of a tree, as shown with the connectivity graph in Fig. 2.1B [36]. The pelvis link is the global root link of the tree. It is connected via a virtual 6-DoF joint to the ground. This connection expresses the fact that the robot has a "floating base," i.e. the base link moves in 3D space like a free rigid body [164]. The root link then branches out into the two legs and the torso. The torso itself represents a local root that branches out into the two arms and the head. The arms terminate with the hands that branch out into the fingers (not shown in the skeleton figure, Fig. 2.1A). The figure shows the so-called *spanning tree* of a humanoid robot. It signifies the case when only the global root link is connected to the ground. The usual situation, however, is that one or more of the other links are also connected. With a quiet stance, for example, the two feet connect to the ground via temporary *contact joints*. In this way, closed loops are formed within the chain. Due to this reason, the kinematic chain of a humanoid robot is characterized as a structurally varying one [107].

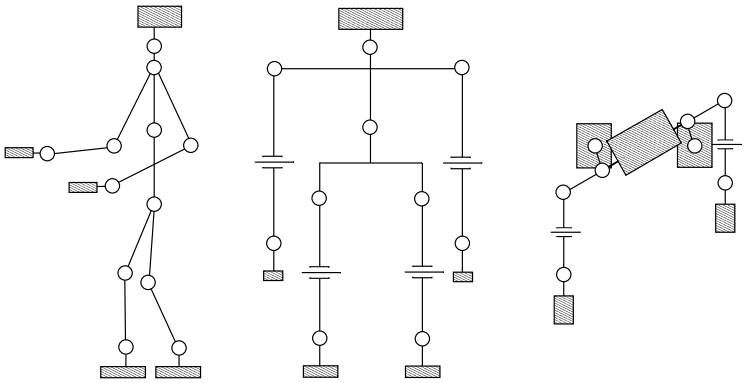

FIGURE 2.2 Planar models in the three planes. (A) Left: sagittal plane. (B) Middle: lateral plane. (C) Right: transverse plane.

In Fig. 2.1A, three reference frames are shown that play special roles. Frame $\{W\}$ is attached to the fixed link (the ground) and hence represents the inertial frame. Frame $\{B\}$ is fixed at the base (root) link. Frame $\{T\}$ is attached to the torso link; it acts as the "local" root for the arm and head branches. There is also another class of coordinate frames with a special role: those attached to the terminal links in the tree structure (the end links). Indeed, the robot interacts with the environment mainly via its end links. For instance, the fingers need to be controlled for grasping and manipulating small objects. The hands, considered as the end links of the arm branches, likewise need to be controlled for fetching/placing small objects and for manipulating larger objects. The feet, on the other hand, are controlled for locomotion. Finally, the head is controlled for an appropriate gaze. In this work, the main focus is on locomotion and arm manipulation, with special emphasis on motion analysis and control of the feet (F) and hands (H). The respective coordinate frames will be denoted as $\{e_j\}$, $e \in \{F,\ H\},\ j \in \{r, l\}$, "$r$" and "$l$" standing for "right" and "left."

Planar models can be quite helpful in motion analysis and motion generation. Three basic planes adopted from human anatomy are used to devise such planar models, as shown in Fig. 2.1A. First, the sagittal plane (S) is an x-z plane perpendicular to the ground. It passes through the "head" and the "spinal cord" and divides the body into left and right (sinister and dexter) portions. Second, the lateral plane (L), also called a frontal or coronal plane, is an y-z plane perpendicular to the ground. It divides the body into back and front (dorsal and ventral, or posterior and anterior) portions. Finally, the transverse plane (T), also referred to as a cross-section, is an x-y horizontal plane parallel to the ground. It separates the head from the feet (or the superior from the inferior). A 3D motion pattern, observed e.g. while walking or while balancing in response to an external disturbance, is often decomposed into motion patterns within the planes. Planar models on the three planes are shown in Fig. 2.2. In the sagittal plane, all joints are 1-DoF rotational joints. In the frontal and transverse planes, on the other hand, the elbow and knee joints are represented as translational joints since the arm and leg lengths appear to vary. This feature is used in motion analysis and generation via simplified models, such as the inverted linear pendulum model described in Chapter 4.

2.3 FORWARD AND INVERSE KINEMATIC PROBLEMS

A rigid-body position/orientation in 3D space, henceforth referred to as *6D position*, is represented as an element of the special Euclidean group $SE(3) = \Re^3 \times SO(3)$. Note that the parametrization of the 3D rotation group $SO(3)$ is not unique [148]. In robotics, most used are 3×3 rotation matrices and the minimal parametrization via Euler angles, as well as Euler parameters (the unit quaternion) [169]. When parametrized by a rotation matrix, for example, bold font notation $\boldsymbol{X} \in SE(3)$ will be used to denote a 4×4 homogeneous matrix (see e.g. [104]). A trailing subscript denotes the link frame, a leading superscript the reference frame. For instance, ${}^{B}\boldsymbol{X}_{F_j}$ expresses the 6D position of the feet w.r.t. the base frame, ${}^{T}\boldsymbol{X}_{H_j}$ gives the 6D position of the hands w.r.t. the torso frame. Furthermore, it is important to note that the elements $\boldsymbol{X}$ signify a coordinate transformation (also called a *rigid-body motion*) between two frames. As an example, ${}^{B}\boldsymbol{X}_{T}$ denotes the transformation from the arm local-root (or torso) frame $\{T\}$ to the base frame $\{B\}$. This transformation is used to express the 6D position of the hand ${}^{T}\boldsymbol{X}_{H_j}$ in base coordinates, i.e.

$$\boldsymbol{X}_{H_j} = {}^{B}\boldsymbol{X}_{T}\,{}^{T}\boldsymbol{X}_{H_j}. \tag{2.1}$$

The two rigid-body motion components, ${}^{B}\boldsymbol{X}_{T}$ and ${}^{T}\boldsymbol{X}_{H_j}$, are obtained from the trunk/limb joint angle data sets via the *forward kinematics functions* for the branches. Denote by $\boldsymbol{\theta}_{e_j} \in \Re^{n_{e_j}}$ the joint angle vector w.r.t. the local-root link, where n_{e_j} stands for the number of joints in the limb. The *forward kinematics problem* is defined as

$${}^{r}\mathcal{X}_{e_j} = {}^{r}\boldsymbol{\varphi}(\boldsymbol{\theta}_{e_j}),$$

where ${}^{r}\boldsymbol{\varphi}(\boldsymbol{\theta}_{e_j})$ is the forward kinematics function and ${}^{r}\mathcal{X}_{e_j} \in WS_r \subset \Re^6$ denotes the 6D position of the end link w.r.t. reference frame $\{r\}$. The 6D position is expressed in *local coordinates*, obtained by employing the minimal parametrization of $SO(3)$ (and hence of $SE(3)$). Local coordinates are used since the minimal parametrization cannot be global [148]. Care is then needed to deal with singularities of the chosen set of Euler angles. WS_r denotes the *workspace* of the limb w.r.t. reference frame $\{r\}$. The derivation of the forward kinematics function for serial-link manipulators is straightforward; see e.g. [104].

Robot tasks are defined w.r.t. the inertial frame embedded into the environment (frame $\{W\}$ in Fig. 2.1A). Then, the 6D position of the controlled bodies (the end links of the limbs) has to be expressed in the inertial frame. This is done via the rigid-body motion transform ${}^{W}\boldsymbol{X}_{B}$. For example, to obtain the 6D position of the hands in $\{W\}$, premultiply (2.1) from the left by ${}^{W}\boldsymbol{X}_{B}$, i.e.

$${}^{W}\boldsymbol{X}_{H_j} = {}^{W}\boldsymbol{X}_{B}\,{}^{B}\boldsymbol{X}_{T}\,{}^{T}\boldsymbol{X}_{H_j}.$$

When the robot is in a single or double stance posture, the rigid-body motion component ${}^{W}\boldsymbol{X}_{B}$ is calculated from the leg(s) joint angle data. When the robot is airborne or tipping over, the base-link localization sensor data obtained via sensor fusion (e.g. from a LIDAR, an IMU, and/or stereo vision) can be used.

For control purposes, the *inverse kinematics problem* has to be solved. That is, "given the 6D position of a specific link (e.g. an end link) in the kinematic chain w.r.t. the branch root frame

or the inertial frame, find the respective joint angles". This problem may have a closed-form solution. An example with humanoid robot HUBO is presented in [126]. Whenever a closed-form solution cannot be found, numerical solutions using differential kinematic relationships have to be used. A closed-form solution is preferable since the differential inverse kinematics have inherent problems, as will be clarified in what follows.

2.4 DIFFERENTIAL KINEMATICS

Differential kinematics are essential for instantaneous motion analysis and motion generation. Since the robot links are represented as rigid bodies, the instantaneous motion of the robot is uniquely characterized by the motion rates in the joints, under the assumption that the instantaneous motion of the base link is known. Information about the motion rates is derived from sensors in the joints (e.g. optical encoders). Using this information, it is possible to calculate the instantaneous motion of any link of interest, e.g. that of the end links. Proper assessment of the instantaneous motion state is essential for activities like walking, manipulating objects with the hands or whole-body reconfiguration to avoid collisions. Further on, differential kinematics also play an important role in control. Task space–based kinematic feedforward/feedback control schemes [136] make use of control commands specified in terms of instantaneous motion of the end links of interest. These control commands are then transformed into joint motion control commands via (inverse) differential kinematics. The differential kinematics are as well essential in deriving dynamical models, wherein both first-order (velocity level) and second-order (acceleration level) relationships are taken into consideration.

2.4.1 Twist, Spatial Velocity, and Spatial Transform

The instantaneous motion of a link is completely characterized by the velocity of a characteristic point (point P) on the link and the angular velocity of the link. These are vector quantities of well-known geometrical origin that are subjected to vector operations such as inner and outer product. In what follows, the following coordinate form representation will be employed: $\boldsymbol{v}_P \in \Re^3$ and $\boldsymbol{\omega} \in \Re^3$, for velocity and angular velocity, respectively. The above vector operations will also be represented in coordinate form. Given two coordinate vectors $\boldsymbol{a},\ \boldsymbol{b} \in \Re^3$, their inner and outer products are denoted as $\boldsymbol{a}^T\boldsymbol{b}$ and $[\boldsymbol{a}^\times]\boldsymbol{b}$, respectively. If $\boldsymbol{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$, then

$$[\boldsymbol{a}^\times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

is the respective outer-product operator, represented as a skew-symmetric matrix. Note that skew-symmetric matrices are characterized by the relation $[\boldsymbol{a}^\times] = -[\boldsymbol{a}^\times]^T$.

Further on, the velocity and the skew-symmetric representation of the angular velocity, $[\boldsymbol{\omega}^\times] \in so(3)$, constitute an element of $se(3)$, the infinitesimal generator of the special Euclidean group $SE(3)$. This element can be parametrized by a 6D vector, $\mathcal{V} \in \Re^6$, composed

of two components representing the velocity and the angular velocity vectors. Given a 6D position in homogeneous coordinates, $\boldsymbol{X}$, describing the motion of a body w.r.t. the inertial frame, the elements of $\mathcal{V}$ can be extracted from $\dot{\boldsymbol{X}}\boldsymbol{X}^{-1}$ or $\boldsymbol{X}^{-1}\dot{\boldsymbol{X}}$. In the former case, the instantaneous motion is described in inertial coordinates and referred to as the *spatial velocity*. In the latter case, the instantaneous motion is described in body coordinates; it will be referred to as the *body velocity*. Sometimes, it might be preferable to express the velocity in a way independent of the particular choice of the characteristic point P, i.e. in the form of a *vector field*. Denote $\mathcal{V} = \begin{bmatrix} \boldsymbol{v}_O^T & \boldsymbol{\omega}^T \end{bmatrix}^T \in \Re^6$. With this parametrization, $\boldsymbol{v}_O$ is interpreted as the velocity of a (fictitious) point on the link that coincides instantaneously with the origin O of an arbitrarily chosen fixed coordinate frame; $\mathcal{V}$, on the other hand, can be interpreted as an operator that, given a point on the link, extracts its velocity in spatial coordinates, i.e.

$$\mathcal{V}(P) \equiv \mathcal{V}_P = \begin{bmatrix} \boldsymbol{v}_P^T & \boldsymbol{\omega}^T \end{bmatrix}^T,$$

where

$$\boldsymbol{v}_P = \boldsymbol{v}_O - [\boldsymbol{r}_{\overleftarrow{PO}}^{\times}]\boldsymbol{\omega}. \tag{2.2}$$

Here $\boldsymbol{r}_{\overleftarrow{PO}}$ denotes the vector pointing from O to P, as apparent from the subscript with the over-arrow notation. The spatial velocity, along with other "spatial" quantities like spatial transform, spatial force, and spatial inertia, constitute the elements of "spatial algebra" [35]. This notation has been widely accepted due to its compactness. Historically, 6D velocity/angular velocity vector pairs (also called bivectors) appeared first in the theory of screws under the name *twist* [9]. A twist can be equated with a spatial velocity vector [104,22]; henceforth these two terms will be used interchangeably. It should be pointed out that implementations based on the spatial notation require great care because of the nonuniform dimensions of the bivectors [31,30].

The instantaneous translation and rotation of the link can be expressed in Cartesian coordinates. Six basis vectors can be defined that constitute a *Plücker coordinate system*, as known from the theory of screws; $\mathcal{V}_P$ then represents the Plücker coordinates of the link spatial velocity for point P. The notation introduced for rigid-body motion objects applies. For instance, the spatial velocity of the hands w.r.t. the base coordinate frame will be expressed as ${}^B\mathcal{V}_{H_j}$. Furthermore, for the purposes of analysis and control it is often required to express the spatial velocity of a point (e.g. point P) on a given link represented in a given coordinate frame (e.g. base frame $\{B\}$), w.r.t. a different coordinate frame (e.g. world frame $\{W\}$). This is accomplished with the following relation:

$${}^W\mathcal{V}_P = {}^W\mathbb{R}_B\, {}^B\mathcal{V}_P,$$

where

$${}^W\mathbb{R}_B = \begin{bmatrix} {}^W\boldsymbol{R}_B & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & {}^W\boldsymbol{R}_B \end{bmatrix} \in \Re^{6\times 6}. \tag{2.3}$$

The leading superscript identifies the reference frame. Notation $\boldsymbol{0}_3$ stands for the 3×3 null matrix, ${}^W\boldsymbol{R}_B \in \Re^{3\times 3}$ denotes the rotation matrix that transforms vectors from the base frame to the world frame, and $\mathbb{R}_B$ will be referred to as the *spatial rotation transform*.

Another operation that is frequently required is the following: given a spatial velocity at point O, find the spatial velocity at another point, P. To this end, the following relation can be employed:

$$ {}^{B}\mathcal{V}_{P} = {}^{B}\mathbb{T}_{\overleftarrow{PO}}\, {}^{B}\mathcal{V}_{O}, \tag{2.4} $$

where

$$ {}^{B}\mathbb{T}_{\overleftarrow{PO}} = \begin{bmatrix} \boldsymbol{E}_3 & -[{}^{B}\boldsymbol{r}^{\times}_{\overleftarrow{PO}}] \\ \boldsymbol{0}_3 & \boldsymbol{E}_3 \end{bmatrix} \in \Re^{6\times 6}. \tag{2.5} $$

Here $\boldsymbol{E}_3$ stands for the 3×3 identity matrix.[1] The validity of (2.2) can be confirmed from the above relation; $\mathbb{T}$ will be referred to as the *spatial translation transform*. The role of this transform is to account for the contribution of the angular velocity of coordinate frame $\{B\}$ to the linear velocity at point P. Note that the translation transform does not change the coordinate frame, as apparent from the same leading superscripts in (2.4). In the particular case when all quantities are represented w.r.t. the world coordinate frame, the leading superscripts will be omitted.

Consecutive spatial translation and rotation can be applied to a twist acting at some given point on the body and expressed in a given coordinate frame to obtain the twist acting at some different point and expressed in a different coordinate frame, as follows:

$$ \begin{aligned} {}^{W}\mathcal{V}_{P} &= {}^{W}\mathbb{R}_{B}\, {}^{B}\mathbb{T}_{\overleftarrow{PO}}\, {}^{B}\mathcal{V}_{O} \\ &= {}^{W}\mathbb{X}_{B_{\overleftarrow{PO}}}\, {}^{B}\mathcal{V}_{P}. \end{aligned} \tag{2.6} $$

The combined spatial transform, $\mathbb{X} : \Re^6 \to \Re^6$, is expressed in this case as

$$ \begin{aligned} {}^{W}\mathbb{X}_{B_{\overleftarrow{PO}}} &= \begin{bmatrix} {}^{W}\boldsymbol{R}_{B} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & {}^{W}\boldsymbol{R}_{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_3 & -[{}^{B}\boldsymbol{r}^{\times}_{\overleftarrow{PO}}] \\ \boldsymbol{0}_3 & \boldsymbol{E}_3 \end{bmatrix} \\ &= \begin{bmatrix} {}^{W}\boldsymbol{R}_{B} & -{}^{W}\boldsymbol{R}_{B}[{}^{B}\boldsymbol{r}^{\times}_{\overleftarrow{PO}}] \\ \boldsymbol{0}_3 & {}^{W}\boldsymbol{R}_{B} \end{bmatrix}. \end{aligned} \tag{2.7} $$

In [104], this transform is called the adjoint transformation associated with the rigid body motion $\mathcal{X} \in SE(3)$, denoted as $\mathrm{Ad}_{\mathcal{X}}$. In [35], the term "Plücker transform" is used to express a connection with the theory of screws. Note that the inverse of the spatial transform,

$$ \begin{aligned} \mathbb{X}^{-1} &= \mathbb{T}^{-1}\mathbb{R}^{-1} \\ &= \begin{bmatrix} \boldsymbol{E} & [\boldsymbol{r}^{\times}] \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}^{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{T} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}^{T} & [\boldsymbol{r}^{\times}]\boldsymbol{R}^{T} \\ \boldsymbol{0} & \boldsymbol{R}^{T} \end{bmatrix}, \end{aligned} \tag{2.8} $$

can be used to exchange the coordinate frames. In the above example, ${}^{W}\mathbb{X}_{B}^{-1} \equiv {}^{B}\mathbb{X}_{W}$.

[1] The subscripts in $\boldsymbol{0}_3$ and $\boldsymbol{E}_3$ will be dropped when there is no ambiguity.

2.4.2 Forward Differential Kinematic Relations

The *forward kinematics problem* for a branch of the kinematic chain (e.g. arm or leg), is formulated as follows: "Given the joint angles and the motion rates in the joints, find the end-link spatial velocity." The instantaneous motion of the end link is expressed in the coordinate frame at the local root of the respective branch. The motion depends on the motion rate in each kinematic joint.

Jacobian Matrix

First, consider a real robot that comprises only single-DoF (rotational) joints. To keep the notation simple, for the time being assume a "generic" branch (or limb) of the kinematic chain composed of n joints. The joint angles are the elements of vector $\boldsymbol{\theta} \in \Re^n$. This vector specifies the *limb configuration*. For a given limb configuration $\boldsymbol{\theta}$, the end-link spatial velocity is determined as a linear combination of the joint motion rates $\dot{\theta}_i$, i.e.

$$\mathcal{V}_n = \sum_{i=1}^{n} \mathcal{V}_i, \;\; \mathcal{V}_i = \mathcal{J}_i(\boldsymbol{\theta})\dot{\theta}_i. \tag{2.9}$$

Here $\mathcal{J}_i(\boldsymbol{\theta})$ denotes the end-link spatial velocity obtained when the joint rate in the ith joint is set at 1 rad/s and all the other joints are locked. This vector can be determined from the following geometrical relation:

$$\mathcal{J}_i(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{e}_i(\boldsymbol{\theta}) \times \Delta\boldsymbol{r}_i(\boldsymbol{\theta}) \\ \boldsymbol{e}_i(\boldsymbol{\theta}) \end{bmatrix}. \tag{2.10}$$

Here $\boldsymbol{e}_i(\boldsymbol{\theta})$ denotes the unit vector along the axis of rotation, $\Delta\boldsymbol{r}_i(\boldsymbol{\theta}) = \boldsymbol{r}_E(\boldsymbol{\theta}) - \boldsymbol{r}_i(\boldsymbol{\theta})$ stands for the position vector of the characteristic point on the end link w.r.t. the link frame chosen, e.g. according to the Denavit and Hartenberg notation [26]. In Fig. 2.3, an example is shown wherein the above vectors are given for the first axis of rotation ($i = 1$) of the right arm. The torso reference frame for the right arm, $\{T_R\}$, is obtained by a translation of the torso root frame $\{T\}$. Formula (2.9) can be used to calculate the spatial velocity of any link from the given branch. Note that when calculating spatial velocity $\mathcal{V}_k$ of Link k, $k < n$, all spatial velocities $\mathcal{V}_j$, $k < j < n$, are zero.

Usually, relation (2.9) is represented in a compact form as follows:

$$\mathcal{V}_n = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{2.11}$$

where matrix $\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} \mathcal{J}_1 & \mathcal{J}_2 & \ldots & \mathcal{J}_n \end{bmatrix} \in \Re^{6\times n}$ denotes the *Jacobian matrix* of the limb. Eq. (2.11) represents the solution to the forward kinematics problem at the velocity level. As apparent from the equation, the solution is unique for the given branch configuration. The forward kinematic problem plays an important role in motion analysis and motion control, especially in feedback kinematic control [136].

Multi-DoF Joint Models

In motion analysis, planning, and simulation, quite often models with multi-DoF kinematic joints are employed, like spherical (S), universal (U), and the 6-DoF rigid-body (RB)

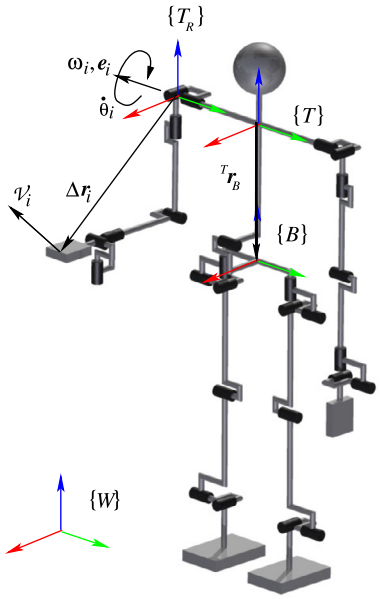

FIGURE 2.3 End-link spatial velocity $\mathcal{J}_i = [(\boldsymbol{e}_i \times \boldsymbol{r}_i)^T \quad \boldsymbol{e}_i^T]^T$ is obtained with joint rate $\dot{\theta}_i = 1$ rad/s. Vector $\boldsymbol{e}_i$ signifies the joint axis of rotation. The position $\boldsymbol{r}_i$ of the characteristic point on the end link is determined w.r.t. reference frame $\{T_R\}$, obtained by translating the common root frame for the arms, $\{T\}$, to a suitably chosen point on the joint axis, e.g. according to the Denavit and Hartenberg notation [26].

joint connecting the base link to the ground. Then, to express the differential kinematic relations between two neighboring links, Link i and its predecessor Link $p(i)$, the following joint model is used. Denote by $\boldsymbol{\vartheta}_i \in \Re^{\eta_i}$ the joint coordinate vector, where η_i is the joint DoF number. The joint DoF is determined as $\eta_i = 6 - c_i$, where c_i is the number of constraints imposed by the joint. The body velocity of Link i is

$$^i\mathcal{V} = {}^i\mathbb{X}_{p(i)}\, {}^{p(i)}\mathcal{V} + {}^i\mathbb{B}_m(\boldsymbol{\vartheta}_i)\dot{\boldsymbol{\vartheta}}_i. \tag{2.12}$$

Here $\mathbb{B}_m(\boldsymbol{\vartheta}_i) \in \Re^{6\times\eta_i}$ is composed of basis vectors that determine the possible motion directions across the joint. In general, the basis vectors depend on the *joint configuration* $\boldsymbol{\vartheta}_i$. When represented in Link i coordinates though, i.e. as in the above equation, $\mathbb{B}_m$ is constant. For the joints mentioned above, we have

$$^i\mathbb{B}_m(\boldsymbol{\vartheta}_S) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad ^i\mathbb{B}_m(\boldsymbol{\vartheta}_U) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad ^i\mathbb{B}_m(\boldsymbol{\vartheta}_{RB}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.13}$$

Models of other types of joints are described in [35], Table 4.1. The above notations imply that rotations in 3D are defined w.r.t. the (body-)fixed axes. Another possibility is to employ rotations w.r.t. the relative coordinate axes. In this case, the joint velocity transform $\mathbb{B}(\boldsymbol{\vartheta})$ is not constant anymore and the underlying expressions are not so simple. The joint model, however, completely matches the physical system, where equivalent spherical joint motion, for instance, is realized by compound 3R joints. Readers interested in detailed analysis of joint models are referred to [73].

Using (2.12), the end-link spatial velocity can be represented in the torso coordinate frame as ${}^T\mathcal{V} = \sum_i^n \left({}^T\mathbb{X}_i {}^i\mathcal{V}\right)$.

Parametrization of Instantaneous Rotation

For the purpose of instantaneous motion analysis it is sometimes necessary to express the end-link spatial velocity as time derivative of a physically meaningful quantity. Note that the spatial velocity $\mathcal{V}$ does not comply with this requirement since the angular velocity vector component does not have an integral. This problem is alleviated by involving a transformation that relates the angular velocity vector to the time differential of the chosen local $SO(3)$ parametrization. For example, assume a minimal parametrization, by employing the ZYX set of Euler angles, $(\phi,\ \theta,\ \psi)$, defined w.r.t. the inertial frame $\{W\}$. Under this parametrization, the angular velocity is represented as

$$\boldsymbol{\omega} = \boldsymbol{A}_{ZYX}(\phi,\ \theta)\begin{bmatrix}\dot{\phi} & \dot{\theta} & \dot{\psi}\end{bmatrix}^T, \tag{2.14}$$

$$\boldsymbol{A}_{ZYX}(\phi,\ \theta) = \begin{bmatrix} 0 & -\sin\phi & \cos\phi\cos\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 1 & 0 & -\sin\theta \end{bmatrix}.$$

The 6D position of the end-link is then expressed by $\mathcal{X} \in \Re^6$ (cf. Section 2.3), while its first-order time differential assumes the form

$$\dot{\mathcal{X}} = \mathbb{X}_{ZYX}(\phi,\ \theta)\mathcal{V} = \mathbb{X}_{ZYX}(\phi,\ \theta)\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{2.15}$$

where $\mathbb{X}_{ZYX}(\phi,\ \theta) = \mathrm{diag}\left[\boldsymbol{E}_3 \quad \boldsymbol{A}_{ZYX}^{-1}(\phi,\ \theta)\right] \in \Re^{6\times 6}$. It is important to note that under the above parametrization, there are specific limb configurations s.t. $\det \boldsymbol{A}_{ZYX} = \cos\theta = 0$, and hence, the inverse transform $\boldsymbol{A}_{ZYX}^{-1}(\phi,\ \theta)$ does not exist. This is a well-known problem inherent to the minimal parametrization of $SO(3)$, via Euler angles. The problem can be alleviated by employing another type of parametrization, e.g. Euler parameters (the unit quaternion) [169]. The same problem exists for the spherical and rigid-body joint models that are based on relative rotations.

By introducing a local parametrization of $so(3)$, as above, the Jacobian matrix can be expressed as the partial derivative of the forward kinematic function, i.e.

$$\boldsymbol{J}_a(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}.$$

Here $\boldsymbol{J}_a$ and $\boldsymbol{J}$ are referred to as the *analytical* and *geometrical* Jacobians, respectively. Note that $\boldsymbol{J}_a \neq \boldsymbol{J}$. For example, under the parametrization used in (2.15), $\boldsymbol{J}_a = \mathbb{X}_{ZYX}\boldsymbol{J}$. Thus, it can

be expected that the use of the geometrical Jacobian in kinematic, kinetostatic, and dynamic analysis yields simpler notations. In fact, it would be more appropriate to refer to $\boldsymbol{J}$ as a joint-to-spatial velocity transform, rather than as a (geometrical) "Jacobian."

2.4.3 Inverse Differential Kinematic Relations

Kinematic control is quite often used with humanoid robots. This type of control is based on differential kinematic relations for the forward and inverse kinematics problems. The later is defined as follows: "Given the joint angles and the end-link spatial velocity, find the motion rates in the joints." In order to find a solution in a straightforward manner, the following two conditions have to be satisfied:

1. the Jacobian matrix at branch configuration $\boldsymbol{\theta}$ should be of full rank;
2. the number of joints of the branch should be equal to the DoF of the end link.

These conditions imply that the inverse of the Jacobian matrix exists. For example, the maximum DoF of each end link of the limb branches (hands and feet) is six. Therefore, according to the above conditions, $\text{rank}\boldsymbol{J}(\boldsymbol{\theta}) = 6$ and $n = 6$. For finger branches used in point-contact grasps, the conditions imply $\text{rank}\boldsymbol{J}(\boldsymbol{\theta}) = 3$ and hence, $n = 3$. Henceforth, without loss of generality, the former (general) case will be considered.

When the conditions are satisfied, solving (2.11) for the joint rates yields the following solution to the inverse kinematics problem:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}\mathcal{V}. \tag{2.16}$$

A branch configuration yielding a full-rank Jacobian is called a *nonsingular configuration*. A branch with a number of joints that conforms to the second condition is called a *kinematically nonredundant* branch.

Whenever any of the above two conditions cannot be met, the inverse problem needs to be handled with care. Indeed, there are special branch configurations, $\boldsymbol{\theta}_s$, where the Jacobian matrix loses rank: $\text{rank}\boldsymbol{J}(\boldsymbol{\theta}_s) < 6$. Such configurations are called *singular*. The branch can attain a singular configuration irrespective of the number of its joints. Further on, when the branch comprises more joints than the DoF of its end link ($n > 6$), then (2.11) is underdetermined. This implies the existence of an infinite set of inverse kinematics solutions for the joint rates [147]. In this case, the branch is referred to as a *kinematically redundant* branch. Singular configurations and kinematically redundant branches will be discussed further in detail in the following subsections.

The second-order (acceleration level) differential kinematics for the forward problem are obtained through differentiating (2.11) w.r.t. time, i.e.

$$\dot{\mathcal{V}} = \boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{2.17}$$

The solution to the respective inverse problem is then represented as

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}\left(\dot{\mathcal{V}} - \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}\right). \tag{2.18}$$

The forward and inverse differential relationships examined so far were "generic," derived w.r.t. a suitably chosen coordinate frame at the local root of the branch. Referring to Fig. 2.3, the hand spatial velocities are conveniently expressed in the common arm frame $\{T\}$ attached at the torso. We introduce now a coordinate frame-dependent notation for the first-order differential relationship (2.11). We have

$$^{T}\mathcal{V}_{H_j} = \boldsymbol{J}(\boldsymbol{\theta}_{H_j})\dot{\boldsymbol{\theta}}_{H_j}. \tag{2.19}$$

The differential relationships for the feet, on the other hand, are more conveniently represented in the base frame $\{B\}$. This is expressed as

$$^{B}\mathcal{V}_{F_j} = \boldsymbol{J}(\boldsymbol{\theta}_{F_j})\dot{\boldsymbol{\theta}}_{F_j}. \tag{2.20}$$

Further on, as already noted, for the purposes of analysis and control it is often required to express all these relationships in the same frame, e.g. in the base frame $\{B\}$ or the inertial frame $\{W\}$. Using rotational operator (2.3), it is easy to obtain the hand spatial velocities in base coordinates as

$$^{B}\mathcal{V}_{H_j} = {}^{B}\mathcal{V}_{T} + {}^{B}\mathbb{R}_{T}\,{}^{T}\mathcal{V}_{H_j} = {}^{B}\mathcal{V}_{T} + {}^{B}\mathbb{R}_{T}\,\boldsymbol{J}(\boldsymbol{\theta}_{H_j})\dot{\boldsymbol{\theta}}_{H_j}, \tag{2.21}$$

where $^{B}\mathcal{V}_{T}$ stands for the spatial velocity of the torso frame. This velocity is a function of the joint angles/rates in the torso joints. Likewise, the spatial velocity of the hand can be obtained in inertial coordinates as

$$^{W}\mathcal{V}_{H_j} = {}^{W}\mathcal{V}_{B} + {}^{W}\mathbb{R}_{B}\,{}^{B}\mathcal{V}_{T} + {}^{W}\mathbb{R}_{T}\,{}^{T}\mathcal{V}_{H_j} = {}^{W}\mathcal{V}_{B} + {}^{W}\mathbb{R}_{B}\,{}^{B}\mathcal{V}_{T} + {}^{W}\mathbb{R}_{T}\,\boldsymbol{J}(\boldsymbol{\theta}_{H_j})\dot{\boldsymbol{\theta}}_{H_j}, \tag{2.22}$$

where $^{W}\mathcal{V}_{B}$ denotes the spatial velocity of the base frame. This velocity can be derived from the leg joint angles and rates, when the robot is standing or walking. In the special cases when the humanoid robot is jumping or running, both legs loose contact with the ground. The base velocity is then determined via the IMU (gyro) sensor of the robot [54], if needed.

2.5 DIFFERENTIAL KINEMATICS AT SINGULAR CONFIGURATIONS

There are certain limb configurations where the end link looses mobility, i.e. the ability of instantaneous motion in one or more directions. Fig. 2.4A shows a robot standing upright with stretched legs and arms hanging at the sides of the body. This posture, though commonly used by humans, is usually avoided with a humanoid robot. The reason is that all four limbs are in singular configurations w.r.t. their root frames ($\{T\}$ and $\{B\}$ for arms and legs, respectively). Indeed, since the arms are fully extended, the hands cannot move in the downward direction w.r.t. the $\{T\}$ frame. Similarly, since the legs are stretched, the $\{B\}$ frame cannot be moved in the upward direction. These singular configurations of the arms/legs are called *elbow/knee singularities*, respectively. They are characterized as *unavoidable* singular configurations: there are no alternative nonsingular configurations that would place the end links at the same locations, at the workspace boundaries of each limb. As apparent from the example,

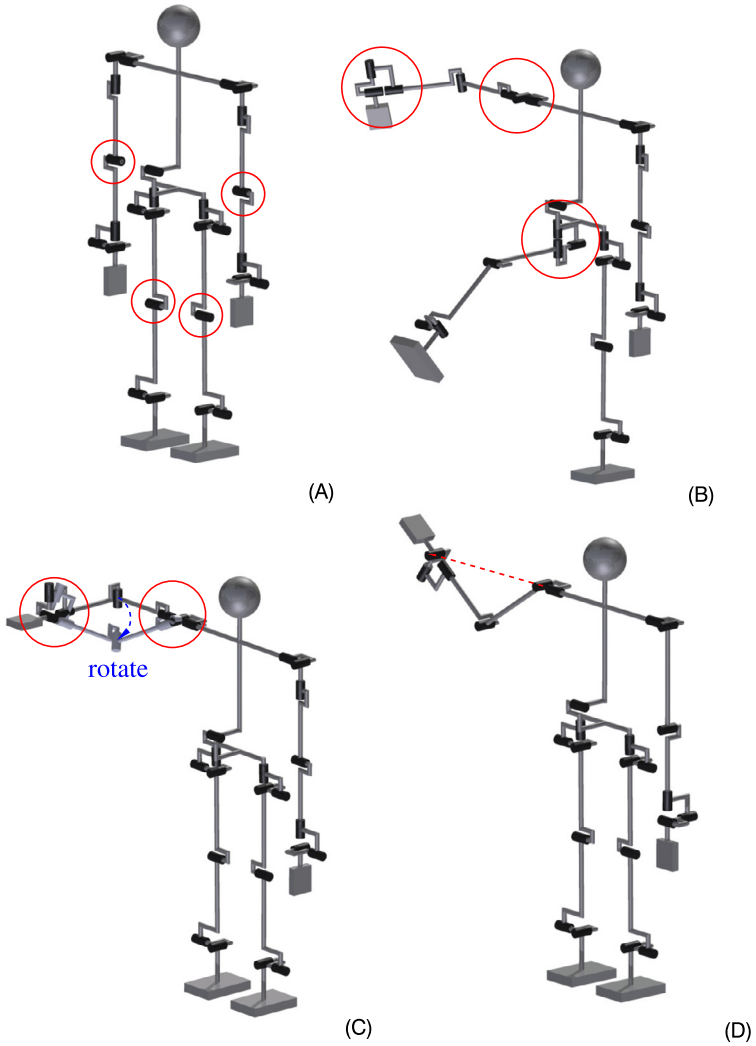

FIGURE 2.4 Singular configurations. (A) Elbow/knee singularities in both arms/legs (unavoidable). (B) Wrist/shoulder singularity in 7-DoF right arm and hip singularity in right leg. (C) Shoulder singularity in 7-DoF right arm (avoidable). (D) Shoulder singularity in 6-DoF right arm (unavoidable).

unavoidable singular configurations are inherent to both redundant and nonredundant limbs (the arms and legs, respectively).

Another singular configuration of the robot is shown in Fig. 2.4B. In addition to the elbow/knee singularities in the left arm/leg, there are singularities associated with the 3-DoF spherical joint–type subchains in the shoulder and wrist joints of the right arm and the hip joint of the right leg, respectively. As apparent from the figure, two out of the three joints axes in each of these joints are aligned, which leads to mobility loss in one direction. The

singular configuration of the arm is called *wrist/shoulder singularity* [71], that for the leg *hip singularity*.

Fig. 2.4C shows another type of singular configuration for the right arm. The singularity is due to the alignment of the two axes in the shoulder joint, the elbow joint being at 90 degrees. The end link loses mobility in the translational direction of the lower-arm link. This configuration is called *shoulder singularity*. Note that the end link is placed within the workspace; it is not on the boundary. In this case, the self-motion of the arm, i.e. a motion whereby the end link is fixed (cf. Section 2.7.1), yields a transition to a nonsingular configuration, as shown in Fig. 2.4C. Such types of singular configurations are characterized as *avoidable*.

A singular configuration for a kinematically nonredundant arm is shown in Fig. 2.4D. It occurs when the wrist center is placed on the axis of the first (root) joint at the shoulder. The translational mobility of the end link is then constrained within the arm plane. This type of configuration is referred to as the *shoulder singularity*. Confusion with the shoulder singularity of the redundant arm should be avoided.

The singular configurations discussed so far are characterized by loss of end-link mobility in a single direction. As already noted, there are singular configurations when the end link can lose mobility in more than one direction. This happens, for example, when the hip and knee singularities in the leg or the wrist/shoulder and elbow singularities in the arm occur simultaneously. Other combinations, as well as other types of singular configurations do exist. Readers interested in in-depth analysis are referred to [71].

When a limb is in a singular configuration, the respective Jacobian matrix becomes rank-deficient, reflecting the loss of mobility. In Fig. 2.4A for example, all four limb Jacobians are rank-deficient; $\text{rank}\boldsymbol{J}_k(\boldsymbol{\theta}_s) = 5,\ k \in \{H_r, H_l, F_r, F_l\}$. The rank is five since each limb has lost mobility in one direction. Multiple singularities, e.g. an instantaneous elbow and wrist singularity, lead to a further decrease in the rank of the Jacobian. Rank deficiency implies that the inverse kinematics solutions cannot be found with the formulas introduced in the previous subsections. The problem can be alleviated in a straightforward manner by avoiding singular configurations, e.g. by using self-motion (kinematically redundant limb) and/or by proper motion planning.

On the other hand, it is important to note that singular configurations can be useful. Indeed, humans use such configurations to resist external forces with minimal load in the joints. For example, during normal walk the support leg is almost fully stretched. This is a configuration in the vicinity of the knee singularity. With this configuration, the ground reaction force imposes a minimal load on the knee joint. Another example is shown in Fig. 2.5: when pushing a heavy object, the arms are almost fully stretched (elbow singularity), as is the pushing leg (knee singularity). Not much of the research so far has addressed the singularity issue: the stretched-knee walking problem for humanoids has been identified in [137,121] and tackled later on in [72,101,56,152,55,46,79,47]. Improved humanoid designs that can circumvent the problematic bent-knee gait have been demonstrated with WABIAN-2/LL, WABIAN-2R, HRP-4C, and other humanoid robots. The use of upper-limb singular configurations is discussed in [70,3].

There are two main approaches to deal with the singularity problem: algebraic and geometric. The algebraic approach is based on the least-squares method for obtaining an approximate solution with a lack of mobility: minimize $\|\boldsymbol{J}\dot{\boldsymbol{\theta}} - \hat{\mathcal{V}}\|^2$, where $\hat{\mathcal{V}}$ is the desired spatial velocity of the limb. At this point it is important to note that the singularity problem should

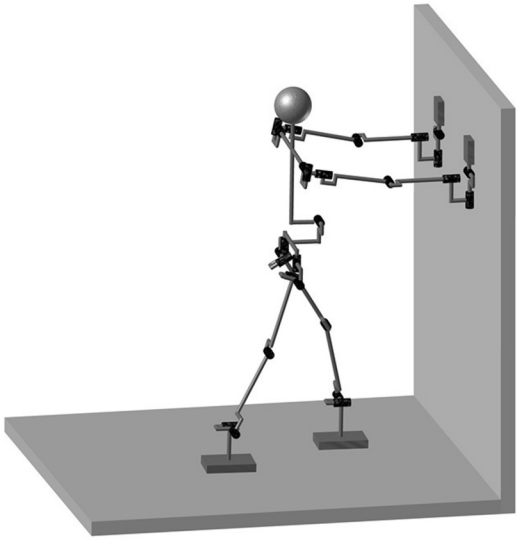

FIGURE 2.5 Pushing with almost singular configurations in the arms and the pushing leg.

not be considered only a pointwise problem. In other words, when the limb is within the vicinity of (but not exactly at) a singular configuration, the Jacobian matrix is ill-conditioned. Then, a high-norm joint velocity will be obtained as a solution to the inverse kinematics problem. This is highly undesirable and therefore, pure least-squares minimization does not work. The problem is alleviated by augmenting the objective with a "damping" term for the joint velocity norm: minimize $\|\boldsymbol{J}\dot{\boldsymbol{\theta}} - \hat{\mathcal{V}}\|^2 + \alpha^2\|\dot{\boldsymbol{\theta}}\|^2$, where α is a weighting factor for the damping. The solution is

$$\dot{\boldsymbol{\theta}} = (\boldsymbol{J}^T\boldsymbol{J} + \alpha^2\boldsymbol{E}_n)^{-1}\boldsymbol{J}^T\hat{\mathcal{V}}. \tag{2.23}$$

The method is known as the *Levenberg–Marquardt* or *Damped Least-Squares* method [78]. Its application in the field of robotics has been pioneered in [106,158]. The method has also been applied to solve the inverse kinematics problem of humanoids when their motion is constrained by multiple tasks. There are, however, a few problems associated with the damping factor α, e.g. the errors it introduces into the solution, its unintuitive nature, and the difficulty of determination. Details will be given in Section 2.8. An attempt to tackle some of these problems has been presented in [149].

Next, consider the geometric approach. The *Singularity-Consistent* (SC) method [111,114] is constraint-based, wherein the differential inverse kinematics are represented in the form of an autonomous dynamical system. The end link is required to follow the desired path (the constraint) exactly, even if it arrives at or passes through a *singular point* in the workspace. Such a point represents the spatial position of the end link obtained as the image of a singular configuration of the limb under the forward kinematics map. A path passing through a singular point is called a *singular path*. Following a singular path is possible since the magnitude of the desired spatial velocity along the path can be set appropriately to comply with the decreased

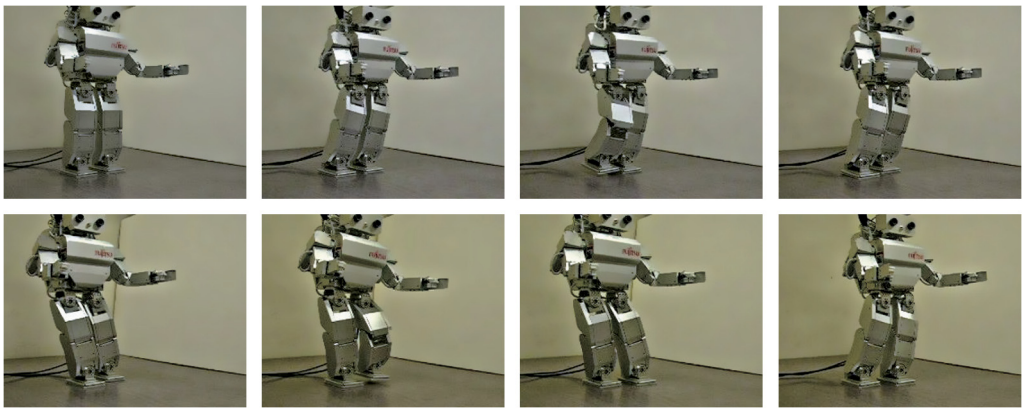

FIGURE 2.6 Implementation of the Singularity-Consistent inverse kinematics approach: stretched-knee walking [152].

mobility of the end link at the singular point. The differential kinematic relations are derived as follows. First, consider a nonredundant limb $n = 6$. The forward kinematic relation (2.11) is rewritten as

$$\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \hat{\mathcal{S}}(\boldsymbol{\theta})\dot{s} = \mathbf{0}. \tag{2.24}$$

The rationale behind this notation is that, according to Chasles' theorem [104], the path constraint can be regarded as a virtual screw motion that closes the kinematic chain. In terms of instantaneous motion, the closure is represented by the normalized twist, $\hat{\mathcal{S}}(\boldsymbol{\theta})$. The scalar $\dot{s}$ denotes the joint rate in the virtual screw joint. It is assumed that $\dot{s}$ is unknown as are the rates in the limb joints. Hence, the above equation is underdetermined. The general solution can be written as

$$\begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{s} \end{bmatrix} = b\boldsymbol{n}(\boldsymbol{\theta}) = b \begin{bmatrix} \{\mathrm{adj}\,\boldsymbol{J}(\boldsymbol{\theta})\}\hat{\mathcal{S}}(\boldsymbol{\theta}) \\ \det \boldsymbol{J}(\boldsymbol{\theta}) \end{bmatrix}, \tag{2.25}$$

where $\boldsymbol{n}(\boldsymbol{\theta})$ denotes a vector spanning the null space of the *column-augmented Jacobian*, $\begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta})^T & -\hat{\mathcal{S}}^T \end{bmatrix}^T$. Scalar b can assume an arbitrary value to scale the joint motion rates in an appropriate way. The notation adj$(\circ)$ denotes the adjoint matrix. With $b = 1/\det \boldsymbol{J}(\boldsymbol{\theta})$, solution (2.16) is obtained. To arrive at a singular point on the workspace boundary along a constrained direction (unavoidable singularity, e.g. the elbow or knee singularity), set b as a constant within a suitably chosen vicinity of the singularity, e.g. determined from the maximum joint speed [113]. This implies that the end link approaches the singular point with a speed proportional to the determinant. This type of motion is referred to as a *natural motion* [114]. It is apparent that the end link's speed will be zero upon arrival at the singular point and, hence, consistent with the singularity constraint. Erect walking patterns (static walk) based on this notation are reported in [56] and [152]. Snapshots from the latter realization are shown in Fig. 2.6.

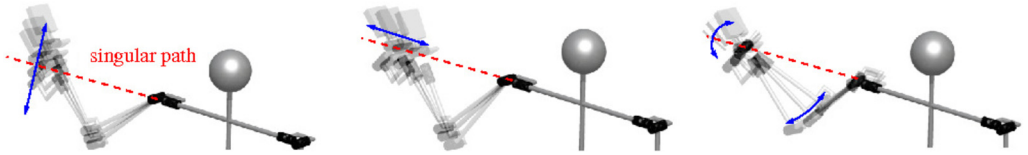

FIGURE 2.7 Implementation of the Singularity-Consistent inverse kinematics approach: three motion patterns at the shoulder singularity of a nonredundant limb. Left (A): motion through the singular path. Middle (B): motion along the singular path. Right (C): rotation around the singular path [156].

In addition to arrival at the singularity, as shown in Video 2.5-1 [98], the SC method also ensures departure from and motion through a singular point. Motion through a singular point on the workspace boundary results in reversal of motion along the path: the end link "reflects" from the boundary, with nonzero joint velocity. This implies a limb reconfiguration, as shown in Video 2.5-2 [99] (i.e. the elbow/knee joint angle changes sign).

Further on, the end-link motion direction along the desired path at the singular point may coincide with the unconstrained motion direction. From the algebraic viewpoint, this means that the linear system, though rank-deficient, will be consistent. Hence, the end link can approach, pass, and depart from the singular point with any speed. In this case, the above differential relations cannot be used since the vector field (2.25) vanishes. This problem can be alleviated via an appropriate coordinate transform: the end-link differential relationships are then expressed w.r.t. a rotating coordinate frame, one of the axes being always aligned with the singular (constrained) direction. Interested readers are referred to [153], where an implementation with a kinematically redundant limb is also explained.

At the (avoidable) shoulder singularity of a nonredundant limb, on the other hand, the SC method can be applied to control instantaneous motions resulting in three motion patterns [156]. These are obtained with the first three joints (the positioning subchain) of a 6R limb, as shown in Fig. 2.7. The dashed (red) line represents the singular path at the shoulder singularity. Whenever the end link is on that path, motion is constrained in the direction parallel to the arm-plane normal. Motion along the unconstrained directions is shown in Figs. 2.7A and B. In Fig. 2.7A, the end link moves through the singularity in the direction transverse to the singular path. In Fig. 2.7B, the end link moves in the direction parallel to the singular path, whereby the singularity will not be escaped. Note that the end link may reach the workspace boundary along the singular path, thus arriving at a double shoulder-elbow singular configuration. Finally, Fig. 2.7C displays a self-motion pattern resulting from a commanded end-link velocity having a nonzero component in the constrained direction (parallel to the arm-plane normal). The joint at the shoulder base rotates the limb and hence the arm plane, until the commanded end-link velocity component along the constrained direction is nullified. Thereafter, the end link will leave the singularity along the unconstrained directions.

As already mentioned, the inverse kinematics solution (2.25) is in the form of an autonomous dynamical system. In [51], it was shown that the nonlinear dynamical system approach can also be used to create attractors for the learning process of a humanoid robot, the so-called "dynamic movement primitives" (DMPs). Recently, DMPs have been implemented in a number of motion generation algorithms.

2.6 MANIPULABILITY ELLIPSOID

From the differential kinematics relation (2.11), it is apparent that the ability of the end link to move instantaneously along a given spatial (rigid-body motion) direction will depend on the current limb configuration. In particular, as already clarified, at a singular configuration the ability to move along the singular directions becomes zero, and hence, mobility is lost in these directions. To facilitate instantaneous motion analysis and control, it is quite desirable to quantify the mobility in a given direction, at any given configuration. This can be done via *Singular-Value Decomposition* (SVD) [42,147,90] of the Jacobian matrix. For the general case of an n-DoF kinematically redundant limb, we have

$$\boldsymbol{J}(\boldsymbol{\theta}) = \boldsymbol{U}(\boldsymbol{\theta})\boldsymbol{\Sigma}(\boldsymbol{\theta})\boldsymbol{V}(\boldsymbol{\theta})^T, \tag{2.26}$$

where $\boldsymbol{U}(\boldsymbol{\theta}) \in \Re^{6\times 6}$ and $\boldsymbol{V}(\boldsymbol{\theta}) \in \Re^{n\times n}$ are orthonormal matrices and

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \left[\text{diag}\{\sigma_1(\boldsymbol{\theta}),\ \sigma_2(\boldsymbol{\theta}), ..., \sigma_6(\boldsymbol{\theta})\} \quad \middle| \quad \boldsymbol{0}\right] \in \Re^{6\times n}. \tag{2.27}$$

Here $\sigma_1 \geq \sigma_2 \geq, ..., \geq \sigma_6 \geq 0$ are the singular values of the Jacobian. The columns of matrix $\boldsymbol{U}(\boldsymbol{\theta})$, $\boldsymbol{u}_i,\ i = 1, ..., 6$, provide a basis for the instantaneous motion space of the end link at the given limb configuration. At a nonsingular limb configuration, all singular values are positive. At a singular configuration of corank $6-\rho$ $(\rho = \text{rank}\boldsymbol{J})$, $6-\rho$ of the singular values become zeros, i.e. $\sigma_1 \geq \sigma_2 \geq, ..., \geq \sigma_\rho > 0,\ \sigma_{\rho+1} = ... = \sigma_6 = 0$. The singular value σ_i quantifies the instantaneous mobility of the end link along the instantaneous motion direction $\boldsymbol{u}_i$. Assuming that the magnitude of the joint rate vector is limited at each limb configuration as $\|\dot{\boldsymbol{\theta}}\| \leq 1$, the highest mobility is along the direction corresponding to the maximum singular value. At a singular configuration of corank 1, $\sigma_{min} = 0$ and the respective direction $\boldsymbol{u}_{min}$ becomes a singular direction. Vectors $\sigma_i \boldsymbol{u}_i$ constitute the principal axis of an ellipsoid—a useful graphic tool for visualizing the instantaneous mobility along each possible motion direction. The dimension of the ellipsoid is determined by the rank of the Jacobian. Fig. 2.8 shows a robot configuration wherein the right arm is at a nonsingular configuration, whereas the left one is at the elbow singularity. The two ellipsoids at the end links visualize the instantaneous translational motion abilities. The ellipsoid for the right arm is 3D (full translational mobility), while that for the left arm is flat (an ellipse). The ellipse lies in a plane parallel to the floor since translational mobility in the vertical direction is nil at the singularity. The ellipsoid-based instantaneous mobility analysis has been introduced in [166]; the ellipsoid is referred to as the *manipulability ellipsoid*.

2.7 KINEMATIC REDUNDANCY

Whenever the limb mobility, determined by the number of limb joints n, exceeds the DoFs of the end link (six), the limb is characterized as kinematically redundant. Some humanoid robots are equipped with kinematically redundant, 7-DoF arms [120,161,57,127,173]. Such robots can control the position of their elbows without affecting thereby the instantaneous

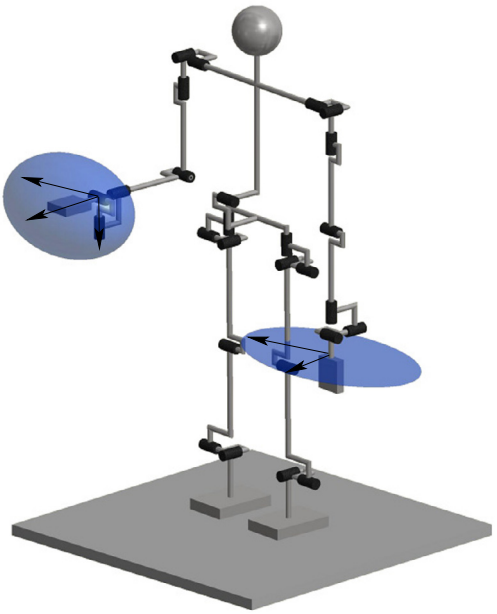

FIGURE 2.8 Manipulability ellipsoid for translational motion. The right arm is in a nonsingular configuration and the respective ellipsoid is 3D, with principal axes $\sigma_1 \boldsymbol{u}_1$, $\sigma_2 \boldsymbol{u}_2$, and $\sigma_3 \boldsymbol{u}_3$. The left arm is at a singular configuration: the downward translational mobility has been lost, and therefore, the manipulability ellipsoid is only 2D. The principal axes are $\sigma_1 \boldsymbol{u}_1$ and $\sigma_2 \boldsymbol{u}_2$.

motion of the hands. Thus, they attain the capability to perform tasks in clustered environments avoiding collisions with their elbows, similar to humans. Also, there are humanoids that comprise 7-DoF legs. With proper control, their gait appears more human-like than that of robots with 6-DoF legs [121,19]. The difference $r = n - 6$ is referred to as the *degree of redundancy* (DoR).

2.7.1 Self-Motion

In contrast to a nonredundant limb, a kinematically redundant limb can move even when its end link is immobilized ($\mathcal{V} = \mathbf{0}$). Such motion is shown in Fig. 2.9 for the arm; the hand remains fixed w.r.t. the arm root frame while the elbow rotates around the line connecting the shoulder and wrist joints. Such type of motion is known as *self-motion*, *internal motion*, or *null motion*.

Self-motion is generated by the joint velocity obtained from the following homogeneous differential relation:

$$\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \mathbf{0}, \quad \dot{\boldsymbol{\theta}} \neq \mathbf{0}. \tag{2.28}$$

Since $n > 6$, the Jacobian is nonsquare ($6 \times n$) and the above equation is characterized as an underdetermined linear system. Hence, there is an infinite set of solutions, each nontrivial

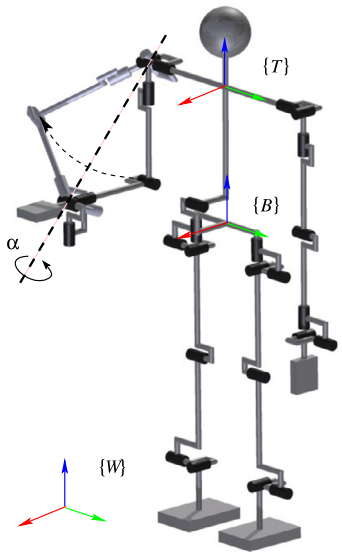

FIGURE 2.9 The self-motion of the arm is shown as a rotation of the arm plane, determined by the upper/lower arm links, around the line connecting the shoulder and wrist joints. The rotation angle α can be associated with parameter b_v in (2.35).

solution representing a self-motion joint velocity:

$$\{\dot{\boldsymbol{\theta}}_h : \dot{\boldsymbol{\theta}} = \boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta}))\dot{\boldsymbol{\theta}}_a,\ \forall \dot{\boldsymbol{\theta}}_a\}. \tag{2.29}$$

Matrix $\boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta})) \in \Re^{n\times n}$, a projection matrix onto the null space of the limb Jacobian, $\mathcal{N}(\boldsymbol{J}(\boldsymbol{\theta}))$, can be expressed as

$$\boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta})) \equiv \left(\boldsymbol{E}_n - \boldsymbol{J}^{\#}(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})\right). \tag{2.30}$$

Matrix $\boldsymbol{J}^{\#} \in \Re^{n\times 6}$ is a generalized inverse of the Jacobian, i.e. a matrix such that $\boldsymbol{J}\boldsymbol{J}^{\#}\boldsymbol{J} = \boldsymbol{J}$. The properties of the generalized inverses are discussed in [14,105]. It is important to note that for a given matrix $\boldsymbol{J}$, there is an infinite number of generalized inverses. Note also that $\boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta}))$ has special properties. They are:

- a singular matrix of rank r (at a nonsingular configuration);
- a symmetric matrix $\boldsymbol{N}^T(\boldsymbol{J}(\boldsymbol{\theta})) = \boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta}))$;
- an idempotent matrix $\boldsymbol{N}^2(\boldsymbol{J}(\boldsymbol{\theta})) = \boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta}))$.

Further on, vector $\dot{\boldsymbol{\theta}}_a$ in (2.29) is an arbitrary joint velocity vector that parametrizes the null space. Because of the rank deficiency of $\boldsymbol{N}$, this parametrization is referred to as the *nonminimal*. A minimal parametrization can be obtained from the SVD of the Jacobian, (2.26), as

$$\dot{\boldsymbol{\theta}}_h = \boldsymbol{V}_r(\boldsymbol{\theta})\boldsymbol{b}. \tag{2.31}$$

Here $\boldsymbol{V}_r(\boldsymbol{\theta}) \in \Re^{n\times r}$ is formed by extracting the last r rows from the orthonormal matrix $\boldsymbol{V}^T(\boldsymbol{\theta})$ in the SVD formula; $\boldsymbol{b} \in \Re^r$ is an arbitrary vector parameter with the dimension of (truncated) joint velocity. The columns of $\boldsymbol{V}_r$ span the null space. Hence, the following alternative representation of the null-space projector can be obtained [69]:

$$\boldsymbol{N}(\boldsymbol{J}(\boldsymbol{\theta})) = \boldsymbol{V}_r \boldsymbol{V}_r^T. \tag{2.32}$$

In the case of a single DoR, $r = 1$, the minimal parametrization assumes the simple form

$$\dot{\boldsymbol{\theta}}_n = b\boldsymbol{n}(\boldsymbol{\theta}), \tag{2.33}$$

where b is an arbitrary scalar with the dimension of joint rate. Vector $\boldsymbol{n} = \begin{bmatrix} n_1 & n_2 & \dots & n_n \end{bmatrix}^T$ is nondimensional; it is the sole nonzero vector spanning the null space. Its components can usually be obtained in closed form, or calculated from cofactors [7,13]: $n_i = (-1)^{i+1} \det \boldsymbol{J}_i$, where $\boldsymbol{J}_i$, $i = 1, 2, .., n$, is derived from the Jacobian matrix by removing the ith column.

It should be noted that a continuous self-motion, $\dot{\boldsymbol{\theta}}_h(t)$, signifies a motion along the integral curves of (2.31). The integral curve for a particular end-link spatial position is defined by $\boldsymbol{f}(\boldsymbol{\theta}) = \text{const}$. The curve will be henceforth referred to as the *self-motion manifold* [18]. The dimension of this manifold equals the rank of the null-space projector $\boldsymbol{N}$ (or $\boldsymbol{V}_r$). The curve can be closed or open, depending on the specified 6D position of the end link [18].

2.7.2 General Solution to the Inverse Kinematics Problem

In the case of a kinematically redundant manipulator, the instantaneous motion relation (2.11) becomes underdetermined. The general solution of an underdetermined linear system can be represented as the sum of a particular solution, $\dot{\boldsymbol{\theta}}_p$, and a solution to the homogeneous equation (2.28), i.e. a self-motion joint velocity $\dot{\boldsymbol{\theta}}_h$ [14]. We have

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{\#}\mathcal{V} + \left(\boldsymbol{E}_n - \boldsymbol{J}(\boldsymbol{\theta})^{\#}\boldsymbol{J}(\boldsymbol{\theta})\right)\dot{\boldsymbol{\theta}}_a. \tag{2.34}$$

Apparently, the particular solution $\dot{\boldsymbol{\theta}}_p = \boldsymbol{J}(\boldsymbol{\theta})^{\#}\mathcal{V}$ is parametrized with the choice of a specific generalized inverse. The choice of $\dot{\boldsymbol{\theta}}_a$, on the other hand, parametrizes the infinite set of solutions. In the case of a single DoR, the above general solution can be rewritten, with the help of (2.33), as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{+}\mathcal{V} + b\boldsymbol{n}(\boldsymbol{\theta}). \tag{2.35}$$

Notation $(\circ)^+$ stands for the Moore–Penrose generalized inverse [14]. This generalized inverse is quite often preferred instead of other generalized inverses because it endows the IK solution for the joint velocity with two desirable properties: minimum norm ($b = 0 \rightarrow \min \|\dot{\boldsymbol{\theta}}\|_2^2$) and orthogonality of the two solution components ($b \neq 0 \rightarrow \dot{\boldsymbol{\theta}}_p \perp \dot{\boldsymbol{\theta}}_h$). The Moore–Penrose generalized inverse is often referred to as the *pseudoinverse*. At a nonsingular configuration, where $\boldsymbol{J}(\boldsymbol{\theta})$ has a full row rank, the (right) pseudoinverse is used. It is expressed with the following formula:

$$\boldsymbol{J}^{+} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}. \tag{2.36}$$

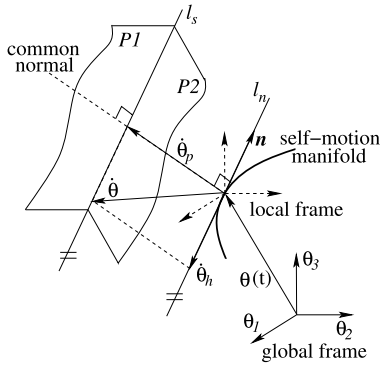

FIGURE 2.10 Geometrical representation of local decomposition of the joint space of a planar 3R redundant manipulator.

The pseudoinverse determines quite a specific behavior of the limb in the joint space. First, note that the induced minimum-norm constraint is of a *nonholonomic* nature, rendering the pseudoinverse-based particular solution nonintegrable. As a consequence, a desired cyclic end-link motion encoded with spatial velocity commands may yield a noncyclic path in the joint space, characterized by a drift [67,85,24]. Second, note that motion can be easily destabilized in the vicinity of a kinematic singularity. This is clearly seen from the following representation of the pseudoinverse, obtained via SVD of the Jacobian (cf. (2.26)):

$$\boldsymbol{J}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{T}, \tag{2.37}$$

where $\boldsymbol{\Sigma}^{+} = \left[\text{diag}\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, ..., \frac{1}{\sigma_6}\} \;\middle|\; \boldsymbol{0}^T\right]^T$. When a singular value σ_i approaches zero, the respective component of the pseudoinverse, and hence of the inverse kinematics solution, tends to infinity, which causes the instability.

Finally, it is important to note that the Jacobian matrix induces a local decomposition in the joint space, whereby the normal and tangential subspaces at the self-motion manifold determine minimum-norm motion and self-motion, respectively, i.e.

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{+}\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}_a + \left(\boldsymbol{E}_n - \boldsymbol{J}(\boldsymbol{\theta})^{+}\boldsymbol{J}(\boldsymbol{\theta})\right)\dot{\boldsymbol{\theta}}_a. \tag{2.38}$$

Consider as an example a planar 3R redundant manipulator, its end tip following a 2D path in workspace. The end-tip velocity $\mathcal{V}$ is along the tangent at the current point on this path. The respective local decomposition of the joint space is presented graphically in Fig. 2.10. Each of the two equations in the forward kinematics relation $\boldsymbol{J}\dot{\boldsymbol{\theta}} = \mathcal{V}$ determines a plane in 3D joint space (planes $P1$ and $P2$). The infinite set of solutions to the inverse kinematics problem is determined by the intersection of the two planes, i.e. line l_s in the figure. Since the DoR is one, (2.35) holds. In the figure, $\dot{\boldsymbol{\theta}}_p = \boldsymbol{J}^{+}\mathcal{V}$ denotes the minimum-norm particular solution, $\dot{\boldsymbol{\theta}}_h = b\boldsymbol{n}$ describes the infinite set of solutions to homogeneous equation $\boldsymbol{J}\dot{\boldsymbol{\theta}} = \boldsymbol{0}$. This set is represented by line l_n. Note that l_n and l_s are parallel. The minimum-norm character of the particular solution $\dot{\boldsymbol{\theta}}_p$ is apparent from its orthogonality w.r.t. the two lines.

The second-order (acceleration level) differential kinematics for the inverse problem can be obtained as the general solution of the underdetermined system (2.17). Making use again either of the nonminimal or the minimal null-space parametrization, two solutions are obtained, respectively, as

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{\#}\left(\dot{\mathcal{V}} - \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}\right) + \left(\boldsymbol{E}_n - \boldsymbol{J}(\boldsymbol{\theta})^{\#}\boldsymbol{J}(\boldsymbol{\theta})\right)\ddot{\boldsymbol{\theta}}_a \tag{2.39}$$

and

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{+}\left(\dot{\mathcal{V}} - \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}\right) + b\boldsymbol{n}(\boldsymbol{\theta}). \tag{2.40}$$

Here $\ddot{\boldsymbol{\theta}}_a$ denotes an arbitrary joint acceleration contributing to the self-motion of the limb. Similarly, the arbitrary scalar b has the physical meaning of joint acceleration magnitude that determines the self-motion. Note, however, that in contrast to the velocity-level solution, a zero input for any of these quantities would not terminate the self-motion. This problem can be alleviated by a careful choice of $\ddot{\boldsymbol{\theta}}_a$ or b, such that the respective inverse kinematics solutions are integrable. Further details will be presented in Section 2.11.3. Alternatively, a joint damping term can be added to damp out the conserved self-motion velocity.

Finally, it is worth noting that the inverse kinematics solution for a 7-DoF limb can be obtained in closed form, as an alternative to the above instantaneous-motion solutions [6, 142,84]. Closed-form solutions may be preferable in some cases since they provide a better accuracy.

2.7.3 Weighted Generalized Inverse

Besides the pseudoinverse, there is another type of generalized inverse that is frequently used in inverse kinematics relations: the *weighted generalized inverse*. In the pioneering work on kinematically redundant manipulators [159], the following generalized inverse was employed:

$$\boldsymbol{J}^{-W} = \boldsymbol{W}^{-1}\boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{T})^{-1}, \tag{2.41}$$

where $\boldsymbol{W} \in \Re^{n\times n}$ is a positive definite weighting matrix. This generalized inverse minimizes locally the weighted norm $\|\dot{\boldsymbol{\theta}}\|_W = \sqrt{\dot{\boldsymbol{\theta}}^T\boldsymbol{W}\dot{\boldsymbol{\theta}}} = \|\boldsymbol{W}^{\frac{1}{2}}\dot{\boldsymbol{\theta}}\|_2$. For example, with a proper constant diagonal $\boldsymbol{W}$, the instantaneous motion in the "heavy-duty" joints can be suppressed w.r.t. that in the rest. This method can also be adopted w.r.t. the end-link velocity components, by minimizing $\|\boldsymbol{v}\|_B$. This implies a variable weighting matrix, $\boldsymbol{W}(\boldsymbol{\theta}) = \boldsymbol{J}_v^T(\boldsymbol{\theta})\boldsymbol{B}\boldsymbol{J}_v(\boldsymbol{\theta})$, matrix $\boldsymbol{J}_v(\boldsymbol{\theta})$ denoting the Jacobian for the end-link velocity. Another example of a variable weighting matrix is $\boldsymbol{W}(\boldsymbol{\theta}) = \boldsymbol{M}(\boldsymbol{\theta})$, where $\boldsymbol{M}(\boldsymbol{\theta})$ is the link inertia matrix. In this case, the weighted generalized inverse is denoted as $\boldsymbol{J}^{-M(\boldsymbol{\theta})}$, since $\boldsymbol{W} = \boldsymbol{M}(\boldsymbol{\theta})$. This generalized inverse plays an important role in inverse dynamics. Details will be discussed in Chapter 4. It should also be apparent that the weighted generalized inverse is a generalization of pseudoinverse (2.36), that is, $\boldsymbol{J}^{+} = \boldsymbol{J}^{-E}$.

2.7.4 Redundancy Resolution via Gradient Projection

With the help of self-motion, it is possible to realize tasks subordinated to the main end-link motion task. Such tasks are referred to as *additional tasks* or *subtasks*. They impose a motion constraint in the joint space that determines the self-motion. Often used additional tasks for a limb are those of the avoidance type; the limb has to avoid joint limits, singular configurations, and collisions with an external obstacle or with other links of the robot (self-collision) [109,143,105].

Joint-Limit Avoidance Subtask

Joint-limit avoidance was tackled in the pioneering work [80]. The method is based on local minimization of the following performance criterion:

$$h(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\theta_i - \theta_{mid_i}}{\theta_{mid_i} - \theta_{max_i}}\right)^2, \tag{2.42}$$

where $\theta_{mid_i} = (\theta_{max_i} + \theta_{min_i})/2$, θ_{max_i} and θ_{min_i} denote the maximum and minimum angular limits in joint i. The minimization is achieved by selecting the arbitrary vector in (2.34) as

$$\dot{\boldsymbol{\theta}}_a = -\beta\frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{2.43}$$

where β is a positive scalar determining the speed of optimization. This method is known as the *gradient projection method*.

Singularity Avoidance Subtask via the Manipulability Measure

Assume a nonsingular limb configuration, $\sigma_i \neq 0, i \in \{1, 2, ..., 6\}$, and denote the product of the singular values as

$$w(\boldsymbol{\theta}) \equiv \sigma_1 \cdot \sigma_2 \cdot, ..., \cdot\sigma_6 = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)}. \tag{2.44}$$

When approaching a singular point in the workspace, scalar $w(\boldsymbol{\theta})$ decreases gradually to become zero at the singularity. In [166], it was suggested to use this scalar as a nondirectional "measure of distance" to the singularity. The maximization of this measure via self-motion leads to the avoidance of (avoidable) singular configurations. As in the above joint-limit avoidance subtask, the maximization is achieved via gradient projection, i.e.

$$\dot{\boldsymbol{\theta}}_a = \beta\frac{\partial w(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{2.45}$$

where scalar $\beta > 0$ determines again the speed of optimization. This scalar can also be used to suppress excessive joint rates, when needed. Further details can be found in [168]. The scalar function $w(\boldsymbol{\theta})$ is referred to as the *manipulability measure* [166].

When using the manipulability as a "distance" measure to singular points, a problem arises due to its nonlinear nature. Appropriate alternatives are the minimum singular value

$\sigma_{min}(\boldsymbol{\theta})$ or the *condition number* $\kappa(\boldsymbol{\theta}) = \sigma_{max}(\boldsymbol{\theta})/\sigma_{min}(\boldsymbol{\theta})$ [68]. Some authors have proposed vector norm–based measures, which however might lead to problems related to the nonuniform dimensions of the two vector components of the spatial end-link velocity [30].

2.7.5 Redundancy Resolution via the Extended Jacobian Technique

The gradient projection method introduced above is based on the nonminimal parametrization of self-motion. An alternative approach is proposed in [8]. It makes use of the fact that the gradient projection term

$$\boldsymbol{N}(\boldsymbol{\theta})\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

that optimizes locally a given scalar function (additional task) $g(\boldsymbol{\theta})$ via self-motion determines $r = n - 6$ independent constraints at a nonsingular configuration. These constraints can be expressed via an r-dimensional vector function $\boldsymbol{g}(\boldsymbol{\theta})$. The relation $\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{0}$ implies that when $g(\boldsymbol{\theta})$ is at its optimum under the forward kinematics constraint, there will be no self-motion.

The additional constraints are represented in terms of instantaneous motion as

$$\boldsymbol{J}_g(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \boldsymbol{0},$$

where $\boldsymbol{J}_g(\boldsymbol{\theta}) = \partial \boldsymbol{g}(\boldsymbol{\theta})/\partial \boldsymbol{\theta} \in \Re^{r\times n}$ is the additional task Jacobian. Adjoining the above additional task constraints to the instantaneous forward kinematics relations yields

$$\begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}) \\ \boldsymbol{J}_g(\boldsymbol{\theta}) \end{bmatrix}\dot{\boldsymbol{\theta}} = \begin{bmatrix} \mathcal{V} \\ \boldsymbol{0} \end{bmatrix}. \tag{2.46}$$

Matrix $\boldsymbol{J}_e(\boldsymbol{\theta}) \equiv \begin{bmatrix} \boldsymbol{J}^T(\boldsymbol{\theta}) & \boldsymbol{J}_g^T(\boldsymbol{\theta}) \end{bmatrix}^T \in \Re^{n\times n}$ is called the *extended Jacobian matrix*. Its inverse can be used to find the unique solution to the instantaneous inverse kinematics problem, provided $\boldsymbol{J}_e(\boldsymbol{\theta})$ is nonsingular.

Further on, the extended Jacobian approach can be generalized by assuming a time-varying additional task, i.e. $\boldsymbol{h}(\boldsymbol{\theta}) = \boldsymbol{\gamma}(t) \in \Re^r$ [141]. In terms of instantaneous motion, the task is

$$\boldsymbol{J}_h(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\gamma}},$$

where $\boldsymbol{J}_h(\boldsymbol{\theta}) = \partial \boldsymbol{h}(\boldsymbol{\theta})/\partial \boldsymbol{\theta} \in \Re^{r\times n}$ is the additional task Jacobian. The extended Jacobian relation becomes

$$\boldsymbol{J}_e(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \begin{bmatrix} \mathcal{V} \\ \dot{\boldsymbol{\gamma}} \end{bmatrix}, \tag{2.47}$$

where $\boldsymbol{J}_e(\boldsymbol{\theta}) \equiv \begin{bmatrix} \boldsymbol{J}^T(\boldsymbol{\theta}) & \boldsymbol{J}_h^T(\boldsymbol{\theta}) \end{bmatrix}^T \in \Re^{n\times n}$ is the extended Jacobian (referred to as the "augmented Jacobian matrix" in [141]).

The drawback of the extended Jacobian approach is the introduction of *algorithmic singularities*. Matrix $\boldsymbol{J}_e(\boldsymbol{\theta})$ can become singular while the manipulator Jacobian $\boldsymbol{J}(\boldsymbol{\theta})$ and the additional task Jacobian $\boldsymbol{J}_h(\boldsymbol{\theta})$ are full row rank. The problem can be alleviated via *kinematic*

decoupling [129]. The components of the vector on the r.h.s. of (2.47) are said to be kinematically decoupled, if the following relations hold:

$$\boldsymbol{J}_h(\boldsymbol{\theta})\boldsymbol{J}_e^{-1}(\boldsymbol{\theta})\begin{bmatrix}\mathcal{V}\\ \mathbf{0}\end{bmatrix}=\mathbf{0},\forall\mathcal{V}$$

and

$$\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}_e^{-1}(\boldsymbol{\theta})\begin{bmatrix}\mathbf{0}\\ \dot{\boldsymbol{\gamma}}\end{bmatrix}=\mathbf{0},\forall\dot{\boldsymbol{\gamma}}.$$

These relations are valid when the Jacobian of the additional constraint is in the form $\boldsymbol{J}_h(\boldsymbol{\theta})=\boldsymbol{Z}(\boldsymbol{\theta})\boldsymbol{W}(\boldsymbol{\theta})$. Matrix $\boldsymbol{Z}(\boldsymbol{\theta})\in\Re^{r\times n}$ is a full row-rank matrix providing a minimal set of basis vectors for null space $\mathcal{N}(\boldsymbol{J})$, i.e. $\boldsymbol{J}\boldsymbol{Z}^T=\mathbf{0}$; $\boldsymbol{W}(\boldsymbol{\theta})$ is a positive-definite weighting matrix. With these definitions, the "asymmetric" weighted generalized inverse

$$\boldsymbol{Z}_W^{\#}(\boldsymbol{\theta})=\boldsymbol{Z}^T(\boldsymbol{\theta})\left(\boldsymbol{Z}(\boldsymbol{\theta})\boldsymbol{W}(\boldsymbol{\theta})\boldsymbol{Z}^T(\boldsymbol{\theta})\right)^{-1}$$

will represent a minimum-parametrization projector onto the null space of the Jacobian, i.e.

$$\{\dot{\boldsymbol{\theta}}:\dot{\boldsymbol{\theta}}_h=\boldsymbol{Z}_W^{\#}(\boldsymbol{\theta})\boldsymbol{b}\}.$$

Vector $\boldsymbol{b}=\boldsymbol{Z}(\boldsymbol{\theta})\boldsymbol{W}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}_a\in\Re^r,\forall\dot{\boldsymbol{\theta}}_a\in\Re^n$. The infinite set of inverse kinematics solutions is determined by

$$\dot{\boldsymbol{\theta}}=\begin{bmatrix}\boldsymbol{J}^{-W}(\boldsymbol{\theta}) & \boldsymbol{Z}_W^{\#}(\boldsymbol{\theta})\end{bmatrix}\begin{bmatrix}\mathcal{V}\\ \boldsymbol{b}\end{bmatrix}. \tag{2.48}$$

The two components of the solution are kinematically decoupled and free of algorithmic singularities, for any $\boldsymbol{W}(\boldsymbol{\theta})$ [122]. These important properties are quite useful not only in inverse kinematics but also in inverse dynamics and passivity control. The method is referred to as the *kinematically decoupled joint-space decomposition* (KD-JSD) method.

2.8 INVERSE KINEMATICS SOLUTION UNDER MULTIPLE TASK CONSTRAINTS

Single-limb differential kinematic relations play an important role for tasks requiring a predetermined posture of the robot. On the other hand, there is a class of tasks wherein the posture of the humanoid is allowed to vary, i.e. a *whole-body* motion is admissible and also desirable. One such task is arm reach, quite often used as a benchmark task; see e.g. [6,59, 81,16]. Thereby, the whole-body posture variation is employed to enlarge the workspace of the arm. With this task, as well as with other similar motion tasks, the motion of the hand(s) is specified w.r.t. the inertial frame $\{W\}$. This means that motions in the joints within the entire kinematic chain will contribute to the hand motion. In the example in Fig. 2.9, the total number of joints becomes $n_{total}=n_{leg}+n_{torso}+n_{arm}=6+1+7=17$. Thus, the degree of

kinematic redundancy is $17 - 6 = 11$. With such a high degree of redundancy, it is possible to realize multiple "additional" tasks.

2.8.1 Motion-Task Constraints

Indeed, besides the hand motion tasks, the robot has to simultaneously perform a number of other motion tasks, e.g. to keep balance while avoiding obstacles, self-collisions, singularities, and joint limits and to follow a moving object visually: the so-called *gaze* task. All these tasks impose constraints on the motion, referred to as the *motion-task constraints*. These constraints are helpful in resolving the kinematic redundancy. Imposing multiple task constraints should be done with great care since the resulting motion may lead to an *overconstrained* state, where no solution to the inverse kinematics problem can be found. Such states are sometimes referred to as a *task conflict*; they occur in fact quite often. Therefore, some means of resolving task conflicts should be embedded into the inverse kinematics solver.

It should be noted that besides motion task constraints, there are also constraints expressed in terms of forces and moments. Such constraints are represented in a subspace dual to that of the motion-task constraints. Force-type constraints may stem from a specific force task (e.g. a joint torque minimization task) or they may stem from the physical contacts occurring when the robot links interfere with objects from the environment. Contact and static force task constraints will be discussed in Sections 2.9 and 3.4, respectively. Force-task constraints within dynamic models, on the other hand, will be treated in Chapter 4.

The problem of determining the most appropriate joint motions under a given set of motion task constraints is nontrivial. A possible approach is to formulate the inverse kinematics problem as a multiobjective optimization problem and to take advantage of readily available numerical optimization methods, e.g. quadratic programming (QP) or differential dynamic programming [38]. An alternative approach is to use an inverse kinematics solution wherein the task-motion constraints are handled within a hierarchical structure derived with the help of null-space projections. This approach was devised for redundant manipulators over three decades ago. The aim was to avoid optimization via general solvers, which was quite time consuming at that time due to the iterative nature of the solver. Recently, it was shown that such solvers can provide optimized solutions in real-time [58]. Both approaches are now under development for use with humanoid robots, on a competitive basis. Proponents of numerical optimization methods refer to the difficulty of incorporating inequality constraints (stemming e.g. from unilateral constraints) with null-space projection methods. Proponents of null-space projection methods, on the other hand, argue that under numerical optimization it is difficult to guarantee control stability. A detailed discussion about these two alternative approaches will be presented in the two following sections.

When tackling motion-task constraint–based inverse kinematics optimization, it is important to distinguish between the following types of constraints:

- link-motion and joint-motion constraints;
- equality- and inequality (unilateral)-type constraints;
- permanently active and temporal constraints;
- high-priority and low-priority constraints.

First, in the case of link-motion constraints, typically the motion of the end links is constrained. For example, the hand moves along a specific path when approaching an object to be grasped, the foot of the swing leg moves along a desired path when taking a step, or the head movement is constrained while tracking an object under a gaze task. Likewise, a link motion-type constraint is imposed by the balance task on the movement of the center of mass (CoM), as will be discussed in detail in Chapter 5. Furthermore, the motion of intermittent links (e.g. related to the spatial motion of the elbow) may be constrained by an obstacle avoidance task.

Second, joint-motion constraints refer to joint angle and rate limits. The joint-limit and the singularity avoidance tasks, discussed in 2.7.4, represent such type of constraints. These tasks are also an example of inequality-type constraints. Obstacle avoidance is another such example, as will be demonstrated in what follows.

Third, to avoid overconstrained states (task conflicts), the number of active tasks should be minimized. This is the so-called *active set* method [83]. Obstacle avoidance is an example of a temporal constraint that should be activated only when an obstacle enters the workspace. Singularity avoidance can also be considered a temporal constraint, to be activated when the measure of distance to the singularity (e.g. manipulability, condition number, minimum singular value) passes a predetermined threshold. Joint limits, on the other hand, are an example of a permanently active constraint.

Finally, introducing priorities between the tasks is helpful to decide whether a temporal constraint should be deactivated to avoid a task conflict. The priority assignment can be either fixed or variable, as will be shown in what follows.

2.8.2 Redundancy Resolution Methods for Multiple Tasks

Motion-task constraints imposed on the motion of specified links are most often resolved via velocity-/acceleration-level inverse kinematics relations. This is because a whole-body motion implies multi-DoF kinematic chains that do not comprise a closed-form solution to the inverse kinematics. As already discussed, kinematic redundancy implies the existence of an infinite set of solutions to the instantaneous inverse kinematics problem. The choice of an appropriate solution is still an open research problem.

Restricted Generalized Inverse and Task Prioritization

Local linear optimization based on the *constrained least-squares* method ([1], Chapter VII) was adopted as a solution in the 1980s [69,45]. The simplest example concerns two tasks: an end-link motion task (the main constraint), expressed with the differential kinematic relation $\boldsymbol{J}_1\dot{\boldsymbol{\theta}} = \mathcal{V}_1$, and an additional motion task (the additional constraint), described by $\boldsymbol{J}_2\dot{\boldsymbol{\theta}} = \mathcal{V}_2$. The objective is to minimize $\|\boldsymbol{J}_2\dot{\boldsymbol{\theta}} - \mathcal{V}_2\|^2$ over all postures, such that the main constraint $\boldsymbol{J}_1\dot{\boldsymbol{\theta}} = \mathcal{V}_1$ is satisfied. The general solution is

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_1^+\mathcal{V}_1 + \overline{\boldsymbol{J}}_2^+\overline{\mathcal{V}}_2 + (\boldsymbol{E} - \boldsymbol{J}_1^+\boldsymbol{J}_1)(\boldsymbol{E} - \overline{\boldsymbol{J}}_2^+\overline{\boldsymbol{J}}_2)\dot{\boldsymbol{\theta}}_a, \tag{2.49}$$

where $\overline{\mathcal{V}}_2 = \mathcal{V}_2 - \boldsymbol{J}_2\boldsymbol{J}_1^+\mathcal{V}_1$. Pseudoinverse $\overline{\boldsymbol{J}}_2^+$ is called a "restricted generalized inverse" ([14], p. 88; see also [97,61]). The overbar notation $\overline{\boldsymbol{J}}_2 = \boldsymbol{J}_2\boldsymbol{N}(\boldsymbol{J}_1)$ is used to denote a *restricted Jacobian* [110]: the range of Jacobian $\boldsymbol{J}_2$ is restricted by the null space of another transform (in

this case, by the null space projector $\boldsymbol{N}(\boldsymbol{J}_1)$). In Chapter 4, this notation will also be applied to other types of transforms, e.g. the joint-space inertia matrix. The above solution introduces an "order of priority" [45,106] among the tasks, the highest priority being assigned to the main task and the second-highest priority to the additional task. Arbitrary joint velocity vector $\dot{\boldsymbol{\theta}}_a$ parametrizes the remaining DoFs, within the intersection $\mathcal{N}(\boldsymbol{J}_1) \cap \mathcal{N}(\boldsymbol{J}_2)$.

It is easy to confirm that when the underdetermined linear system

$$\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \mathcal{V}_1 \\ \mathcal{V}_2 \end{bmatrix} \tag{2.50}$$

has full (row) rank, its pseudoinverse solution coincides with the minimum-norm solution derived from (2.49), i.e. with $\dot{\boldsymbol{\theta}}_a = \boldsymbol{0}$. This means that the above task-prioritization scheme makes sense only when system (2.50) is rank-deficient. But then, either $\boldsymbol{J}_1$ and/or $\boldsymbol{J}_2$ or $\overline{\boldsymbol{J}}_2$ is rank-deficient, and hence, at least one of the (right) pseudoinverses in (2.49) ceases to exist. Therefore, when implementing (2.49), some means of handling singularities should be provided. The rank deficiencies of $\boldsymbol{J}_1$, $\boldsymbol{J}_2$, and $\overline{\boldsymbol{J}}_2$ are referred to as kinematic, *task*, and *algorithmic* singularity [8,110], respectively. An algorithmic singularity indicates linear dependency among the tasks, such that $\text{rank}\begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{J}_2^T \end{bmatrix}^T < \text{rank}\boldsymbol{J}_1 + \text{rank}\boldsymbol{J}_2$. This type of rank deficiency is the essence of a *task conflict*.

Multiple Tasks With Fixed Priorities

With the help of the arbitrary vector parameter $\dot{\boldsymbol{\theta}}_a$ in (2.49), it is possible to extend the hierarchical inverse kinematics structure to cope with more than two subtasks. The following recursive scheme has been proposed for m subtasks [109,115]:

$$\begin{aligned} \dot{\boldsymbol{\theta}}_k &= \dot{\boldsymbol{\theta}}_{k-1} + \overline{\boldsymbol{J}}_k^+ \overline{\mathcal{V}}_k, \\ \overline{\mathcal{V}}_k &= \mathcal{V}_k - \boldsymbol{J}_k \dot{\boldsymbol{\theta}}_{k-1}, \\ \overline{\boldsymbol{J}}_k &= \boldsymbol{J}_k \boldsymbol{N}_{k-1}, \\ \boldsymbol{N}_k &= \prod_{i=1}^{k} (\boldsymbol{E} - \overline{\boldsymbol{J}}_i^+ \overline{\boldsymbol{J}}_i), \\ k &= 1, 2, ..., m, \;\; \dot{\boldsymbol{\theta}}_0 = \boldsymbol{0}, \;\; \boldsymbol{N}_0 = \boldsymbol{E}. \end{aligned} \tag{2.51}$$

This scheme is characterized by the restriction of Jacobian $\boldsymbol{J}_k$ by the intersection of null spaces $\mathcal{N}(\boldsymbol{J}_1) \cap \mathcal{N}(\boldsymbol{J}_2) \cap ... \cap \mathcal{N}(\boldsymbol{J}_{k-1})$.

Another recursive formulation has been presented in [144], i.e.

$$\begin{aligned} \dot{\boldsymbol{\theta}}_k &= \dot{\boldsymbol{\theta}}_{k-1} + \overline{\boldsymbol{J}}_k^{\#} \overline{\mathcal{V}}_k, \\ \overline{\mathcal{V}}_k &= \mathcal{V}_k - \boldsymbol{J}_k \dot{\boldsymbol{\theta}}_{k-1}, \\ \overline{\boldsymbol{J}}_k &= \boldsymbol{J}_k \boldsymbol{N}_{C_{k-1}}, \\ \boldsymbol{N}_{C_{k-1}} &= (\boldsymbol{E} - \boldsymbol{J}_{C_k}^{\#} \boldsymbol{J}_{C_k}), \end{aligned} \tag{2.52}$$

$$\boldsymbol{J}_{C_k} = \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{J}_2^T & .... & \boldsymbol{J}_k^T \end{bmatrix}^T,$$
$$k = 1, 2, ..., m, \;\; \dot{\boldsymbol{\theta}}_0 = \boldsymbol{0}, \;\; \boldsymbol{N}_{C_0} = \boldsymbol{E}.$$

This scheme is characterized by the restriction of Jacobian $\boldsymbol{J}_k$ by the null space $\mathcal{N}(\boldsymbol{J}_{C_k})$. The Lyapunov-based stability analysis in [2] has clarified that, to guarantee stability, task relations in the first scheme should be defined in a more conservative way than those in the second one. As shown in [115], both schemes are characterized by the following: (1) the kth task is of lower priority than the first $k-1$ tasks due to the fixed priority and (2) the kth task cannot be realized in two cases: either due to task singularity, i.e. when the first $k-1$ tasks are all well conditioned but matrix $\boldsymbol{J}_k$ is not, or due to an algorithmic singularity ($\boldsymbol{J}_k$ is well conditioned but the restricted Jacobian $\overline{\boldsymbol{J}}_k$ is not). The latter case means however that the first $k-1$ tasks definitely influence the executability of task k. The two schemes can be easily extended to second-order differential relations [135].

In the two schemes (2.51) and (2.52), the task priorities are fixed. Based on the above analysis, it is straightforward to conclude that the way of allocating priorities to the tasks will be crucial for the performance. In humanoid robotics, fixed priorities have been introduced in [138], where three main priority levels were suggested: highest priority for joint-motion constraints, mid-level priority for link-motion constraints, and lowest priority for postural-variation constraints. Indeed, joint-motion constraints cannot be violated since at the extreme, they become physical constraints, e.g. hitting a joint limit or saturating a joint rate. At the mid-level priority level, link-motion constraints are further hierarchically substructured to account for the need of a higher priority, e.g. for balance (the CoM control subtask) w.r.t. the hand position control. At the lowest level, postural-variation constraints refer to constraints on the remaining DoFs (if any), imposed in joint space, such as trying to pull all joint angles toward their center values, i.e., $\theta_i \rightarrow \theta_{mid_i}$.

A pioneering example of redundancy resolution with hand motion control as the main task and *obstacle avoidance* as a low-priority (additional) task is presented in [89]. It is assumed that the shortest distance d between the obstacle and a specific point on the (intermittent) link closest to the obstacle is available from sensor data. The instantaneous motion of this point, expressed in terms of spatial velocity $\mathcal{V}_2$, is to be controlled for obstacle avoidance. Based on (2.49), the obstacle avoidance solution was derived as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_1^+ \mathcal{V}_1 + \alpha_\eta \overline{\boldsymbol{J}}_2^+ (\alpha_2 \mathcal{V}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^+ \mathcal{V}_1), \tag{2.53}$$

where subscripts "1" and "2" refer to the main (hand motion) and additional (obstacle avoidance) tasks, respectively. Two scalar variables, $\alpha_\eta(d)$ and $\alpha_2(d)$, were introduced to determine the behavior of the manipulator in the vicinity of an obstacle. Three specific distances around the obstacle were determined as the task-abort distance, the unity-gain distance, and the sphere-of-influence distance. The latter is used to activate self-motion via α_η, when approaching the obstacle. Outside the sphere of influence, self-motion can be used for other subtasks, thus avoiding unnecessary overconstraints. Once activated, the rate of self-motion gradually increases, i.e. to ensure an appropriate repelling behavior via velocity $\alpha_2 \mathcal{V}_2$. The value of α_2 increases quadratically when the link comes closer to the obstacle. A collision is imminent when the task-abort distance is reached.

The above example clearly demonstrates a major problem inherent to the multitask null-space projection methods: the *difficulty in dealing with unilateral (inequality-type) constraints*.

The scalar $\alpha_2(d)$ determines in fact a repelling potential that represents a unilateral constraint. Potential functions are typically employed in obstacle avoidance tasks [8,65,41,168], but also in joint-limit and singularity avoidance subtasks. Apparently, expressions (2.42) and (2.44) represent such functions. Note that these potentials remain activated throughout the entire motion, as their gradients are projected onto the null space. In contrast, the potential in the obstacle avoidance example is activated/deactivated via the sphere-of-influence threshold. This approach, known as the *active set* method [83], is preferable in lieu of permanently active potentials, since in the latter case the system may easily get overconstrained. On the other hand, the nature of activation/deactivation is discrete, and therefore, special means for smoothing are needed to avoid discontinuity in the joint velocity solution. In the obstacle-avoidance example, this was achieved via $\alpha_\eta(d)$, defined as a smooth polynomial of d, with boundary values "0" (no self-motion) and "1" (self-motion active).

Variable Task Priorities With Smooth Task Transitions

Another possible approach for dealing with algorithmic singularities (overconstrained system) and the related discontinuity was reported in [115,112]. The main idea is to *dynamically allocate the priority among the tasks*. The method makes use of the fact that when the first $k-1$ tasks have a consistent solution, they can be reordered arbitrarily without affecting the solution. According to the method, a multidimensional task is *fully decomposed* into single-dimensional components. The priority of each component is determined dynamically, at each time step, the best-conditioned task component receiving thereby the highest priority. Less well-conditioned components have lower priority, and ill-conditioned components, if any, have the lowest priority. The latter are damped in a smooth way, in accordance with the idea behind the damped least-squares method described in Section 2.5. Thereby, because of the full task decomposition approach, the time consuming SVD is avoided.

Others have also considered variable task-priority allocation. In [17], the idea of exchanging the priorities among the main task and an obstacle avoidance subtask was proposed. The reasoning is that if a collision becomes inevitable under low-priority obstacle avoidance, then by exchanging the priorities, there will be a possibility to avoid the collision. In [29], it was argued that the priority of unilateral constraints (joint limits typically) does not have to be always fixed at the highest level. With regard to variable-priority tasks, in [63,75,131] methods for smooth task activation/deactivation have been developed.

Algorithmic singularities are inherent to the above prioritization schemes. As already noted, an algorithmic singularity signifies an overconstrained system due to numerous task constraints. Given a desired set of tasks, this is an unavoidable problem. There are two possible approaches to deal with this problem. First, a "damping" or "regularization" term can be added to the solution to suppress the excessive joint velocity; in essence, this is the damped least-squares approach outlined in Section 2.5. Second, the number of tasks can be decreased, e.g. via removal of the "least important" ones. The former method has been criticized by a number of researchers for introducing errors into the solution, disturbing the order of priority, the unintuitive character of the damping factor, and the difficulty of its tuning [116,93,63, 29]. Therefore, the latter method is preferable. Note though that "simple" removal (or deactivation) would imply an abrupt change in system dimensions and hence discontinuity in the solution. The easiest way to alleviate this problem is by introducing a variable scalar function at each level of the hierarchy. The role of this variable is similar to that of α_η in the obstacle

avoidance solution (2.53). As an example, consider the approach proposed in [75]. In the case of two tasks, the priority-based solution discussed above, is modified as follows:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_1^+ \mathcal{V}_1^{int} + \overline{\boldsymbol{J}}_2^+ (\mathcal{V}_2^{int} - \boldsymbol{J}_2 \boldsymbol{J}_1^+ \mathcal{V}_1^{int}), \tag{2.54}$$

where

$$\mathcal{V}_1^{int} = \alpha_1 \mathcal{V}_1 + (1 - \alpha_1) \boldsymbol{J}_1 \boldsymbol{J}_2^+ \alpha_2 \mathcal{V}_2,$$

$$\mathcal{V}_2^{int} = \alpha_2 \mathcal{V}_2 + (1 - \alpha_2) \boldsymbol{J}_2 \boldsymbol{J}_1^+ \alpha_1 \mathcal{V}_1$$

are intermediate values that ensure smooth transitions via appropriate choices of scalars $\alpha_1,\ \alpha_2 \in [0,\ 1]$. It is straightforward to rewrite the scheme for m tasks. The number of scalar functions increases accordingly, i.e. $\alpha_k, k = 1, 2, ..., m$. These functions are called *activation variables*.

Another possible approach is to modify the null-space projectors $\boldsymbol{N}_k$ appearing in (2.51) for smooth task transitions as follows [131,130]:

$$\boldsymbol{E} - \alpha_k \overline{\boldsymbol{J}}_k^+ \overline{\boldsymbol{J}}_k. \tag{2.55}$$

Scalars α_k have the meaning of activation variables, as above. Their definition plays a crucial role in the behavior of the robot and should be done in a task-dependent way. In [130], for instance, exponential functions of the zero-moment point (ZMP) position were involved to ensure "reflexive" balance control of a humanoid. Other examples of activation variables can be found in [17,63].

In [28], the so-called "null space projection shaping" method was proposed for a self-collision avoidance task of m DoFs. Note that self-collision is handled in the same way as obstacle avoidance, i.e. via potential functions, since the constraints are again unilateral [150, 145]. According to the method, first the null space projector (2.30) is recast via SVD as follows:

$$\boldsymbol{N} = (\boldsymbol{E}_n - \boldsymbol{J}^+ \boldsymbol{J}) = (\boldsymbol{E}_n - \boldsymbol{V} \boldsymbol{\Sigma}^+ \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T) = (\boldsymbol{E}_n - \boldsymbol{V} \overline{\boldsymbol{E}} \boldsymbol{V}^T), \tag{2.56}$$

where $\overline{\boldsymbol{E}} = \text{diag}(\boldsymbol{E}_m,\ \boldsymbol{0}_{n-m})$. The m unit diagonal elements in $\overline{\boldsymbol{E}}$ indicate that all task components are active. A task component is deactivated by simply replacing the respective "1" on the diagonal with a "0." For smooth transitions, however, the m diagonal elements are redefined as differentiable scalar functions, having the meaning of activation variables as in the above schemes.

2.8.3 Iterative Optimization Methods

When looking for a solution to the inverse kinematics problem under a set of motion-task constraints, the first thought is to apply a readily available optimization method like linear-quadratic or differential dynamic programming. This approach was first used for animation of human figures [172] whereby the main goal was to decrease the animator's workload. Tasks in 3D space, such as reaching a point with the hand, were formulated as potential functions. Multiple tasks were incorporated then as a sum of weighted potentials. The objective potential function was subjected to joint limits, using a nonlinear programming approach

with linear equality- and inequality-type constraints. This method, however, turns out to be quite cumbersome since weight tuning is empirical. Moreover, with multiple tasks, the system may easily become overconstrained, or it may get stuck at a suboptimal solution.

Most used are convex optimization methods, a subclass of nonlinear programming methods. Especially, the following quadratic programming (QP) task formulation is suitable:

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{G}\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x}, \tag{2.57}$$

subject to

$$\begin{aligned}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{a} &= \boldsymbol{0},\\ \boldsymbol{B}\boldsymbol{x} + \boldsymbol{b} &\leq \boldsymbol{0}.\end{aligned}$$

Usually, the minimization problem is set in the form $0.5\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{c}\|^2$, thus $\boldsymbol{G} = \boldsymbol{C}^T\boldsymbol{C}$ and $\boldsymbol{g} = -\boldsymbol{C}^T\boldsymbol{c}$ [38]. Note that inequality constraints can be handled with this formulation, which is quite suitable, e.g. to ensure the joint-limit constraints.

Introducing a Hierarchical Structure With Fixed Task Priorities

It is desirable to avoid an over-constrained system. One possible approach is to arrange the tasks within the optimization framework in a hierarchical structure [96]. In [23], a recursive prioritization scheme was devised wherein the constrained optimization process at the current priority level remained decoupled from all high-priority processes. Decoupling was achieved via null-space projection operators, in a manner similar to the redundancy resolution methods for multiple tasks. The optimization objective was defined as follows.

For a set of tasks $T_i(\boldsymbol{x}_i),\ \ i = 1, 2, ..., m$, find

$$\begin{gathered}h_i = \min_{\boldsymbol{x}_i \in S_i} T_i(\boldsymbol{x}_i),\\ \text{subject to}\\ T_k(\boldsymbol{x}_i) = h_k, \forall k < i,\end{gathered}$$

where S_i is a nonempty convex set. The tasks were determined as positive semidefinite quadratic forms of linear equality constraints, i.e.

$$T_i(\boldsymbol{x}_i) = \|\boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{b}_i\|^2,$$

for given $\boldsymbol{A}_i,\ \boldsymbol{b}_i$. For example, in the case of instantaneous motion tasks, the following replacements hold: $\boldsymbol{x}_i \to \dot{\boldsymbol{\theta}}_i$, $\boldsymbol{A}_i \to \boldsymbol{J}_i$, $\boldsymbol{b}_i \to \mathcal{V}_i$. The solution to this problem at priority level k is based on appropriate parametrization of null spaces of restricted Jacobians at all preceding $k-1$ levels.

It turns out that a method called *lexicographic optimization* [52] solves the same problem; a finite number of objective functions are to be optimized on a feasible set in a lexicographic order, i.e. low-priority tasks are optimized as far as they do not interfere with the optimization of higher-priority ones. This method was not well known in the robotics community

until recently; it first appeared in relation to a rigid-body contact modeling [146,163]. The lexicographic order is denoted as follows:

$$T_1(\boldsymbol{x}) \succ T_2(\boldsymbol{x}) \succ \ldots \succ T_m(\boldsymbol{x}).$$

This notation signifies hierarchical structuration within the set of tasks.

Variable Task Priorities With Smooth Task Transitions

The importance of the task activation/deactivation feature in combination with reordering the task priorities has been discussed in [91,145,63]. The respective control architecture comprises a low-level controller that ensures a "reflexive" or "reactive"-type task execution with priority determined by the sequence of the tasks defined within a so-called "stack of tasks." The high-level controller makes the decisions about the activation/deactivation (inserting/removing from the stack) of specific tasks and their priority allocation within the stack. In [92,94], the problem of discontinuity in the solution is addressed via activation variables α_k, as already introduced above. These variables are placed on the diagonal of a matrix $\mathbb{A}$ referred to as the *activation matrix*. The inverse kinematics solution is derived in the form

$$\dot{\boldsymbol{\theta}} = \lambda(\mathbb{A}\boldsymbol{J})^{+}\mathbb{A}\mathcal{V}, \tag{2.58}$$

where scalar λ determines the convergence speed of the iterative solution. Further on, it was confirmed that when dealing with the discontinuity of the above pseudoinverse, the damped least-squares method is inappropriate, due to the reasons already discussed. Instead, a new inverse operator was proposed. For the simplest case of two subtasks, the inverse is derived via the following decomposition of the pseudoinverse:

$$\boldsymbol{J}^{+} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix}^{+} = \begin{bmatrix} \boldsymbol{J}_1^{+} & \boldsymbol{J}_2^{+} \end{bmatrix} + \boldsymbol{C}_{12}, \tag{2.59}$$

where $\boldsymbol{C}_{12}$ accounts for the *coupling* between the two subtask Jacobians $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$. Then, the solution (2.58) is rewritten as

$$\dot{\boldsymbol{\theta}} = \lambda \boldsymbol{J}_c^{\oplus\mathbb{A}}\mathcal{V}, \tag{2.60}$$

where $\boldsymbol{J}_c^{\oplus\mathbb{A}}$ is the new inverse having the property of continuity. The new inverse and the respective solution (2.60) were verified with vision-based servoing experiments. No other reports are available so far with regard to humanoids, though.

Introducing Inequality Constraints

The advantage of iterative optimization methods is that inequality constraints can be embedded in a straightforward manner, via *slack variables*. This approach was implemented for humanoid robots [60,59], as follows. Instead of the above tasks T_i, consider the following linear inequalities:

$$\boldsymbol{b}_i^{min} \leq \boldsymbol{A}_i\boldsymbol{x}_i \leq \boldsymbol{b}_i^{max}.$$

The minimization objective is then

$$\min_{\boldsymbol{x}_i \in S_i, \boldsymbol{w}_i} \|\boldsymbol{w}_i\|, \;\; \boldsymbol{b}_i^{min} \leq \boldsymbol{A}_i \boldsymbol{x}_i - \boldsymbol{w}_i \leq \boldsymbol{b}_i^{max}, \tag{2.61}$$

subject to

$$\boldsymbol{b}_k^{min} \leq \boldsymbol{A}_k \boldsymbol{x}_i \leq \boldsymbol{b}_k^{max}, \;\; \forall k < i.$$

The role of the slack variable $\boldsymbol{w}_k$ is to relax the constraints at level k. The norm $\|\boldsymbol{w}_k\|$ can be used as a measure of constraint violation at level k. This violation is propagated to all lower levels.

In these schemes, the particular solutions and the null space basis vectors at each priority level are determined via SVD. The aggregate computational cost of this method, however, is quite high. Recently, alternative approaches have been explored, based on complete orthogonal decomposition [33,34] and QR factorization [58]. Encouraging results that support real-time implementations have been achieved. The QR factorization scheme has been designed to account for the decreasing dimension of the optimization problem when descending through the hierarchical structure. This method was shown to yield the fastest result.

With the computational cost problem being manageable, the remaining problem associated with numerical optimization is the computational stability. Note that in the above schemes, the prioritization is based on restricted Jacobians. As with the redundancy resolution methods for multiple tasks, the inherent algorithmic singularities hinder definitive conclusions about stability. The damped least-squares method was considered in [58] to alleviate the problem. But as already noted, this method suffers from a number of drawbacks. The stability of numerical optimization schemes with task prioritization still remains an open issue.

In a recent work [82], a generalized hierarchical IK algorithm was proposed that claims to avoid the numerical instability problem. The method can handle inequality constraints, as well as fixed and variable task priorities. In the latter case, multiple priority rearrangements can be executed simultaneously with the help of the so-called "generalized null-space projector." It is also claimed that the method is robust w.r.t. both kinematic and algorithmic singularities. Unfortunately, the heavy computational load does not allow for real-time control.

2.8.4 Summary and Discussion

The focus in this section was on motion-task constraints and the respective null-space projections derived through motion-task Jacobians. It was shown that with such projections, a desirable hierarchy among the tasks can be introduced, either fixed or variable. Null-space projections play an important role in the two competing approaches—the classic one, established with studies on kinematically redundant manipulators in the 1980s, and the newer one, based on iterative optimization methods and developed initially to support human figure animation techniques. The desirable features in both approaches are:

- dealing with tasks arranged in a hierarchical structure;
- handling equality- and inequality-type tasks;
- dealing with singularities;
- flexibility through variable priority allocation for the tasks;

- possibility for activation/deactivation of tasks (i.e. the application of the *active set* method [83]);
- ensuring smooth transitions between the tasks;
- stability and passivity of the solution;
- compliance with hard real-time requirements.

Iterative optimization methods are general; they can cope with all types of constraints, including inequality ones. In the past, ensuring real-time performance was problematic. But recently, it was shown that such performance is achievable for frameworks with null space–based task prioritization. The remaining problem is the lack of a proof regarding stability when implemented into robot control.

Not all of the methods discussed here were purely kinematics-based. Dynamical relations need to be taken under consideration when addressing control stability. Asymptotic stability of multitask null space-based redundancy resolution was proven so far in [27,140] only. A related problem is the lack of passivity when using null-space projections, as identified in [133]. In [124], it was shown how to ensure conditional passivity for compliance control.

Switching task priorities and task activation/deactivation is important to avoid an overconstrained system due to task conflicts (or algorithmic singularities). Since these are discrete events, special means have been developed to ensure a smooth solution. In the past, the damped least-squares method was the preferred approach. However, a number of problems arise when implementing the method into hierarchical multitask schemes, as pointed out in [116,93,63,29]:

- the damping factor works against the desirable order of priority;
- the accuracy of the solution decreases not only in the vicinity of a singularity;
- the tuning of the damping factor is empirical and difficult;
- the damping factor is unintuitive and cannot be related to physical parameters of the system.

It is possible to deal with multitask constraints via weighted generalized inverses, instead of null-space projections [172,21,44]. Most researchers agree, however, that this method is not preferable since weight tuning is empirical and lacks rigorousness. Here also, no proof exists with regard to control stability.

The null space–based approaches discussed here can ensure a "reflexive" or "reactive" robot behavior. The nature of this type of behavior is local. Such behavior is considered a necessary but insufficient condition. Global motion planning methods need to be involved as well, as noted e.g. in [63,29,25].

Motion-task constraints only were discussed here. Physical constraints stemming from contacts will be discussed in the following section.

2.9 MOTION CONSTRAINTS THROUGH CONTACTS

A biped humanoid robot moves around by walking, establishing thereby physical contacts between the feet and the ground. Physical contacts also occur when the robot performs manipulation tasks using thereby the hands and fingers to manipulate an object or a tool.

Contacts occurring between one or more of the end links and an object from the environment (the ground, a wall, and so on) are typical cases of physical contacts [107]. Not so common are contacts established between an object and one or more of the robot's intermittent links, e.g. when the robot lies on the ground after a fall [40,66,76,43], while it is sitting in a chair [134,77], or when it is pushing a heavy object with its back [102]. The robot may also contact an object indirectly via a hand-held tool, or when placing an object on a table [20]. Sometimes, a physical contact may occur between the robot links, without involving an object from the environment (e.g. in the case of self-collision).

All these types of physical contacts are modeled via *contact joints*. Modeling and analysis of contact phenomena in multibody systems and robotics have been extensively studied since the 1990s [100,10,154,125,128]. The mobility of the robot is constrained via the contact. Hence, a *constrained multibody system* model becomes appropriate. Research in the area of constrained multibody systems has drawn in fact a lot of attention since the turn of the century [4,12,165, 15,74]. In the field of robotics, the constrained-motion control problem can be traced back to manipulator force [160], hybrid position/force [132], and cooperative multiarm robot control [48,167]; see also [32,171]. Dexterous object manipulation with a multifingered hand is another well-established area of robotics research that refers to constrained motion models. The theoretical base of hand modeling and control, as described in [104] (Chapters 5 and 6), provides a firm foundation for humanoid robot modeling and control, as will be apparent from what follows. In the field of humanoid robots, the importance of motion and force control across contact joints has been recognized e.g. in [128,108,92,139,134].

2.9.1 Contact Joints

A contact joint enforces up to six constraints upon the relative motion between the two bodies in contact. The number of motion constraints enforced depends on the shapes of the bodies and assumptions with regard to friction [100]. These conditions determine the type of the contact joint.

A contact joint with friction constrains the motion in all directions. The focus here will be on *frictionless* contact joints that allow relative motion in one or more directions. Contact joints with friction will be treated in Section 3.3. In the example depicted in Fig. 2.11, the robot stands on the ground and holds a cylindrical object (a rod) with the hands. Contacts occur between the feet and the ground and between the hands and the object. Assuming zero friction at the hand contacts, sliding along and axial rotation around the rod is admissible. The two hand-contact joints can then be characterized as cylindrical joints. The foot contact joints, on the other hand, can be characterized as planar joints since sliding within the ground plane and rotation around the normal is admissible with zero friction. Another often discussed example is object grasping with the finger tips. The contact model includes point-contact joints, without or with friction [104].

The contact joints at the end links will be identified by $k \in \{e_r, e_l\}, e \in \{H, F\}$. The number of motion constraints at each contact joint will be denoted as c_k, $c_k \leq 6$. With $c_k = 6$, the relative motion between the two bodies in contact is fully constrained. With $c_k < 6$, relative motion along $\eta_k = 6 - c_k$ directions is possible. These directions are referred to as *unconstrained*. There are two special types of "contact" joints. A contact joint that enforces constraints along all six directions ($c_k = 6$, $\eta_k = 0$) is called a "welded" joint. Weld-type joints are used to model a

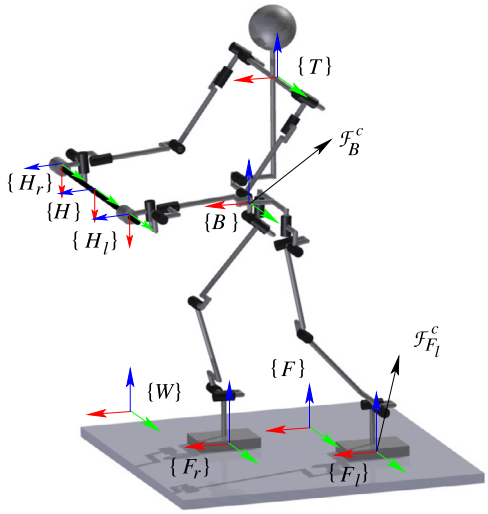

FIGURE 2.11 Closed loops are formed via contact joints at the feet and hands. Contact coordinate frames $\{k\}$, $k \in \{e_r, e_l\}$, $e \in \{H, F\}$ are fixed at the center of pressure (CoP) to the common loop-closure link (floor F and rod H). The z-axes at the feet (shown in blue color) point in a way s.t. the *reaction* force at the contact is always nonnegative. The contact constraints in the vertical direction at the feet are unilateral while those in the angular tangential directions are bilateral, with bounds. All contact constraints at the hands are bilateral.

firm grip of an object or a tool, for example. Next, recall the rigid-body joint introduced in Section 2.4.2. This joint can be characterized as a "contact" joint that does not enforce any constraints, s.t. $c_k = 0,\ \eta_k = 6$.

2.9.2 Contact Coordinate Frames

The constrained/unconstrained motion directions at the contact joint are expressed in an appropriate *contact coordinate frame*. The origin of this frame is fixed at the characteristic contact point, on the site of the loop-closure link. For point contacts, the characteristic point is trivial; for surface or line contacts, the *center of pressure (CoP)* is used as such point [139]. The position of the CoP can be derived through appropriate sensor data, e.g. from force sensing resistors distributed over the soles or from multiaxis force/torque sensors [53]. The z-axis of the contact coordinate frame is along the surface normal of the object that is in contact with the robot link. By convention, the direction of the z-axis is set in a way s.t. the *reaction* force at the contact is always nonnegative [10]. In the case of grasping, for example, such force always results from compression (squeezing the object). On the other hand, the x- and y-axes are chosen by intuition, in dependence with the specific object surface [162]. The commonly accepted contact modeling approach requires also the definition of another coordinate frame. This frame is fixed to the second body taking part in the contact, i.e. to the robot link [100,104, 125].

Consider as an example the contact joints at the hands and the feet displayed in Fig. 2.11. Contact frames $\{e_k\}$ are fixed at the rod and the ground, at the four CoPs. The z_k-coordinate

axes are along the object surface normal, which always denotes a motion constraint direction. In the case of the feet contacts, the z_{F_j}-axes point upward so that the reaction contact force along the normal is always positive. The x_{F_j}- and y_{F_j}-coordinate axes are tangent to the contact surface (the ground). The x_{F_j}-axes are selected to be parallel to the inertial frame x_F-axis. In the case of the hand contacts, the y_{H_j}-axes are selected to be parallel to the object's y_H-axis.

Generally, the surfaces coming into contact may have any shape [100,125]. One should bear in mind, then, that there are cases when the selection of the contact frame axes is not as straightforward as in the present example [37].

2.9.3 Kinematic Models of Frictionless Contact Joints

Denote by $\overline{\mathcal{V}}_k^m \in \Re^{\eta_k}$ the first-order instantaneous motion components along the unconstrained-motion directions at contact joint k. These components determine the contact joint twist, i.e.

$$\mathcal{V}_k = {}^k\mathbb{B}_m \overline{\mathcal{V}}_k^m. \tag{2.62}$$

Here ${}^k\mathbb{B}_m \in \Re^{6\times\eta_k}$ is a transform that comprises orthonormal basis vectors for the twist components in the unconstrained motion directions.[2] There is a complementary transform s.t. ${}^k\mathbb{B}_m \oplus {}^k\mathbb{B}_c = \boldsymbol{E}_6$ ($\oplus$ denotes the direct sum operator):

$$\mathcal{V}_k = {}^k\mathbb{B}_c \overline{\mathcal{V}}_k^c. \tag{2.63}$$

Here $\overline{\mathcal{V}}_k^c$ comprises first-order instantaneous motion components in the constrained motion directions. In the above notations (and throughout this text), the overbar notation signifies a restricted quantity, i.e.

$$\overline{\mathcal{V}}_k^m = \boldsymbol{N}({}^k\mathbb{B}_c)\mathcal{V}_k = {}^k\mathbb{B}_m^T\mathcal{V}_k, \tag{2.64}$$

$$\overline{\mathcal{V}}_k^c = \boldsymbol{N}({}^k\mathbb{B}_m)\mathcal{V}_k = {}^k\mathbb{B}_c^T\mathcal{V}_k. \tag{2.65}$$

These relations imply that

$$\begin{bmatrix} \overline{\mathcal{V}}_k^c \\ \overline{\mathcal{V}}_k^m \end{bmatrix} = \begin{bmatrix} {}^k\mathbb{B}_c^T \\ {}^k\mathbb{B}_m^T \end{bmatrix} \mathcal{V}_k, \quad \overline{\mathcal{V}}_k^c \perp \overline{\mathcal{V}}_k^m. \tag{2.66}$$

Note that in the case of *hard* constraints $\overline{\mathcal{V}}_k^c = \boldsymbol{0}$.

The above transforms are represented in the contact coordinate frame. On the other hand, the motion of objects within the environment that come in contact with the robot links is usually expressed in terms of inertial coordinates. This means that expressions for the transforms in the inertial (world) frame are needed. To this end, use the following relation:

$$\mathbb{B}_{(\circ)}(\boldsymbol{p}_k) = \mathbb{R}_k(\boldsymbol{p}_k)\,{}^k\mathbb{B}_{(\circ)}. \tag{2.67}$$

Here $\mathbb{R}_k$ is the spatial rotation matrix defined in (2.3). Apparently, when expressed in the inertial frame (indicated by a missing lead superscript), $\mathbb{B}_m(\boldsymbol{p}_k)$ and $\mathbb{B}_c(\boldsymbol{p}_k)$ become functions

[2] Recall that such transform was used in the multi-DoF joint models (cf. (2.13)).

of the contact geometry parametrized by the local curvature and torsion of the two surfaces in contact. The respective parameters are collected in the vector parameter $\boldsymbol{p}_k$ [64,100]. This is the reason why $\mathbb{R}_k$ is specified as a function of $\boldsymbol{p}_k$. It should also be noted that the $\mathbb{R}_k(\boldsymbol{p}_k)$ map preserves the complementarity relation between the two transforms.

Examples

In the example in Fig. 2.11, the frictionless cylindrical contact joints at the hands determine

$$ {}^{H_j}\mathbb{B}_m = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \overline{\mathcal{V}}_{H_j}^{m} = \begin{bmatrix} v_y \\ \omega_y \end{bmatrix}. \tag{2.68} $$

The frictionless plane-contact joints at the feet, on the other hand, are modeled with

$$ {}^{F_j}\mathbb{B}_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \overline{\mathcal{V}}_{F_j}^{m} = \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix}. \tag{2.69} $$

The contact models described so far have been expressed in the contact coordinate frame on the environment/object site. To accomplish tasks like walking and object manipulation, the robot has to control the motion (and force) components across the contact joints. Therefore, contact frames at the robot site have to be introduced into the contact model. This will be done in the following section.

2.10 DIFFERENTIAL KINEMATICS OF CHAINS WITH CLOSED LOOPS

Constraints on the instantaneous motion within the kinematic chain are imposed by the controller (*task-based* constraints) and/or via contacts (*physical* constraints). For example, the additional motion constraints used for redundancy resolution in Section 2.7 are task-based constraints. End-link path following, e.g. according to the SC method (cf. Section 2.5), gives another example. The main focus here will be on physical constraints through contacts.

Through the contact joints, one or more closed loops are formed within the robot kinematic chain. When the robot is in a double stance, for example, a closed loop is formed via the ground/feet contacts. The ground is referred to as the *loop-closure link*. When holding an object with the hands, a second closed loop is formed via the object/hands contacts. In this case, there are two loop-closure links: the ground and the object. Each closed loop is formed

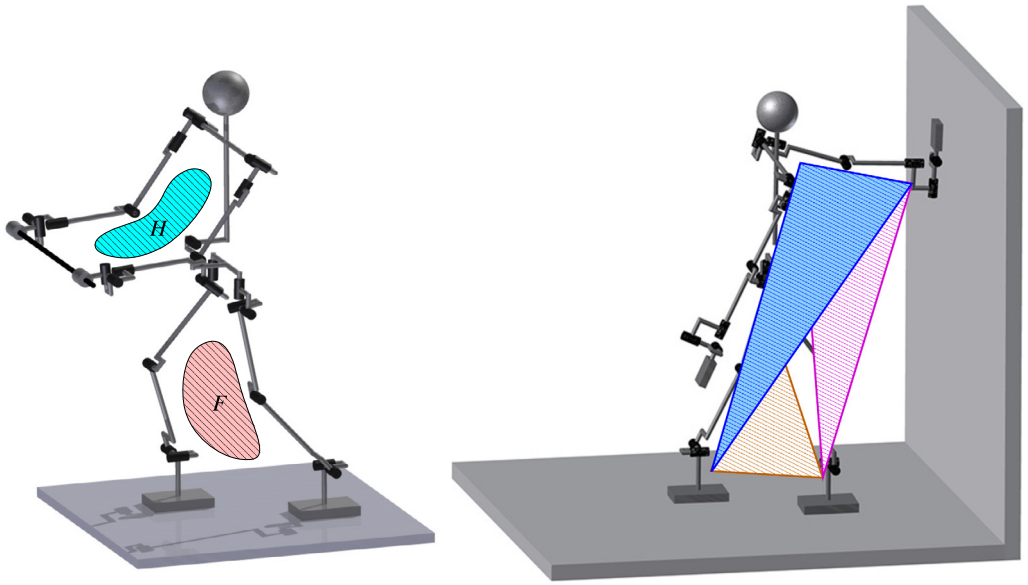

FIGURE 2.12 Closed loops formed via contact joints. Left (A): Two *independent* closed loops are formed with two distinct loop-closure links, floor F and object (rod) H, when the feet rest at the floor and the hands hold an object (a rod). The loops are identified by the respective loop-closure links (floor F and rod H). Each loop comprises two parallel branches composed of the legs and arms, respectively. Right (B): The three colored triangles signify three *interdependent* closed loops formed with three parallel branches: hand/left foot (blue), hand/right foot (red), and foot/foot (yellow). The common loop-closure link is composed of the floor and the wall.

by two parallel branches: the arms and the legs. The two loops are said to be *independent*. This situation is depicted in Fig. 2.12A. With this model, it is possible to design two independent controllers, e.g. a balance controller for the closed loop with the legs and an object manipulation controller for the closed loop with the arms.

Further on, when one of the hands touches a wall while the robot is in a double stance, three closed loops are formed. Note that there is only one common loop-closure link that is composed by the floor and the wall. This link gives rise to interdependent differential motion relations. The three closed loops formed by the three parallel branches, i.e. the two legs and the arm touching the wall, are said to be *interdependent*. Referring to Fig. 2.12B, the three colored triangles and the floor/wall signify the closed loops and the common loop-closure link, F, respectively. Another example where interdependent closed loops are formed is manipulation of an object with the fingers [104]. The fingers represent multiple parallel branches; the grasped object plays the role of the loop-closure link. In this way, multiple interdependent closed loops are formed.

2.10.1 Instantaneous Motion Analysis of Chains With Closed Loops

Closed-loop subchains exhibit differential kinematic relations that reflect the loss of DoF due to the contact constraints within each loop. In this section, the focus will be on the single

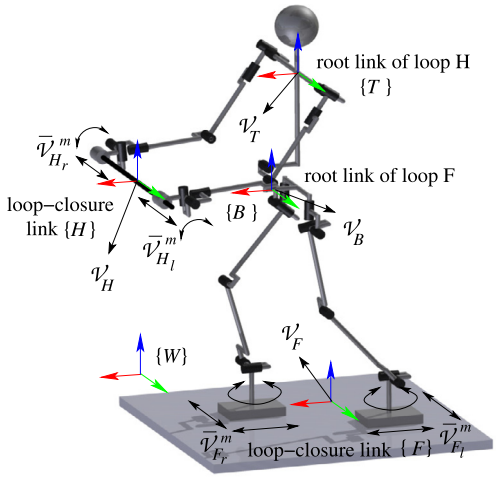

FIGURE 2.13 Humanoid robot with two independent closed loops, F and H, formed by frictionless contact joints at the feet and hands, respectively. The loop-closure link velocities are denoted as $\mathcal{V}_F$ and $\mathcal{V}_H$. The end-link velocities $\overline{\mathcal{V}}^{\,m}_{e_j}$ signify instantaneous motions in the unconstrained directions at each contact joint. Each closed loop is composed of two physical parallel branches connecting the loop-root with the loop-closure link, plus one virtual branch (not shown) formed via a virtual 6-DoF joint between the inertial frame and the root link.

closed-loop case, i.e. closed loop F or H, as shown in Fig. 2.12A. For the purpose of instantaneous motion analysis, frictionless contact joints will be assumed unless otherwise specified. Joints with friction will be treated in Section 3.4. Contact coordinate frames $\{k\},\ \ k \in \{e_r,\ e_l\}$, $e \in \{H, F\}$, are assigned according to the rules clarified in Section 2.9.2. The basis vectors along the constrained and unconstrained motion directions are denoted as ${}^k\mathbb{B}_c \in \Re^{6\times c_k}$ and ${}^k\mathbb{B}_m \in \Re^{6\times \eta_k}$, respectively (cf. Section 2.9.3). In the following derivations, all quantities will be expressed in the inertial frame, and therefore, the leading superscript will be omitted. The basis vectors at the contact joints are transformed to the inertial frame in accordance with (2.67). In this way, it becomes possible to handle curved contact surfaces (nonflat floor/curved object). The root and closure links of the closed loop will be denoted as $\{R\}$ and $\{e\}$, respectively. The closed loops formed by the legs and arms are thus characterized by $e = F,\ R = B$ and $e = H,\ R = T$, respectively (cf. Fig. 2.13). The number of limb joints is denoted as $n_k \geq 6$, $n_e = n_{e_r} + n_{e_l} \geq 12$ standing for the total number of actuated joints in the closed loop. The respective joint variables are collected in the vector $\boldsymbol{\theta}_e = \left[\boldsymbol{\theta}_{e_r}^T \quad \boldsymbol{\theta}_{e_l}^T\right]^T \in \Re^{n_e}$. Furthermore, the closed-loop generalized coordinate vector is denoted as $\boldsymbol{q}_e = (\mathcal{X}_R,\ \boldsymbol{\theta}_e) \in \Re^{6+n_e}$, $\mathcal{X}_R$ standing for the 6D position of the root link determined with the preferred choice of local coordinates (e.g. Euler angles, Euler parameters; cf. Section 2.4.2).

The instantaneous motion relations within the closed loop are determined by the twists of the closure and root links, $\mathcal{V}_e$ and $\mathcal{V}_R$, as well as by the loop joint rates, $\dot{\boldsymbol{\theta}}_e$. Referring to Fig. 2.13, the rod held by the hands moves instantaneously with twist $\mathcal{V}_H$. Also, the general case of a support surface, moving instantaneously with twist $\mathcal{V}_F$, can be addressed with this notation.

Limb Velocities

The instantaneous motion of the contact frame k is determined as

$$\mathcal{V}_k(\mathcal{V}_e, \boldsymbol{p}_k) = \mathbb{T}_{\overleftarrow{ke}}(\boldsymbol{p}_k)\mathcal{V}_e + \mathcal{V}_k'(\boldsymbol{p}_k), \tag{2.70}$$

where $\mathcal{V}_k'(\boldsymbol{p}_k)$ denotes relative instantaneous motion w.r.t. the closure link. Recall that the vector parameter $\boldsymbol{p}_k$ parametrizes the contact joint motion in unison with the contact geometry (cf. (2.67)). The instantaneous motion of end link k, on the other hand, can be written as

$$\mathcal{V}_k(\mathcal{V}_R, \boldsymbol{q}_k) = \mathbb{T}_{\overleftarrow{kR}}(\boldsymbol{q}_k)\mathcal{V}_R + \boldsymbol{J}_R(\boldsymbol{q}_k)\dot{\boldsymbol{\theta}}_k, \tag{2.71}$$

where $\boldsymbol{J}_R(\boldsymbol{q}_k) \in \Re^{6\times n_k}$ denotes the limb Jacobian. The closure of the kinematic loop will be ensured when the end links track the motion of the contact frames on the closure link precisely. We have

$$\mathcal{V}_k(\mathcal{V}_e, \boldsymbol{p}_k) = \mathcal{V}_k(\mathcal{V}_R, \boldsymbol{q}_k).$$

Simplifying assumption: From the above equation one can conclude that motion at a contact joint is parametrized by $n_k + 6 + \dim(\boldsymbol{p}_k)$ number of variables. This means that, in general, the instantaneous motion relations in the closed loop will depend on the time derivatives of the $\boldsymbol{p}_k$ vector parameters [64]. To keep the notation simple, henceforth time-invariant contact geometries only will be assumed. Then, $\mathcal{V}_k'(\boldsymbol{p}_k)$, $\mathbb{T}_{\overleftarrow{ke}}(\boldsymbol{p}_k)$, $\mathbb{B}_c(\boldsymbol{p}_k)$, and $\mathbb{B}_m(\boldsymbol{p}_k)$ can be expressed as functions of $\boldsymbol{q}_k$ instead of $\boldsymbol{p}_k$. The twist at the contact joint will be denoted simply as

$$\mathcal{V}_k \equiv \mathcal{V}_k(\mathcal{V}_R, \boldsymbol{q}_k). \tag{2.72}$$

Further on, (2.71) can be projected along the constrained (c) and mobility (m) (i.e. the unconstrained) directions. To this end, premultiply (2.71) first by $\mathbb{B}_c^T(\boldsymbol{q}_k)$ and then by $\mathbb{B}_m^T(\boldsymbol{q}_k)$, to obtain the system

$$\begin{bmatrix} \mathbb{C}_{cR}^T(\boldsymbol{q}_k) \\ \mathbb{C}_{mR}^T(\boldsymbol{q}_k) \end{bmatrix} \mathcal{V}_R + \begin{bmatrix} \mathcal{J}_{cR}(\boldsymbol{q}_k) \\ \mathcal{J}_{mR}(\boldsymbol{q}_k) \end{bmatrix} \dot{\boldsymbol{\theta}}_k = \begin{bmatrix} \mathbb{B}_c{}^T(\boldsymbol{q}_k) \\ \mathbb{B}_m^T(\boldsymbol{q}_k) \end{bmatrix} \mathcal{V}_k, \tag{2.73}$$

where

$$\mathbb{C}_{cR}^T(\boldsymbol{q}_k) = \mathbb{B}_c^T(\boldsymbol{q}_k)\, \mathbb{T}_{\overleftarrow{kR}}(\boldsymbol{q}_k) \in \Re^{c_k\times 6}, \tag{2.74}$$

$$\mathcal{J}_{cR}(\boldsymbol{q}_k) = \mathbb{B}_c^T(\boldsymbol{q}_k)\boldsymbol{J}_R(\boldsymbol{q}_k) \in \Re^{c_k\times n_k}, \tag{2.75}$$

$$\mathbb{C}_{mR}^T(\boldsymbol{q}_k) = \mathbb{B}_m^T(\boldsymbol{q}_k)\, \mathbb{T}_{\overleftarrow{kR}}(\boldsymbol{q}_k) \in \Re^{\eta_k\times 6},$$

$$\mathcal{J}_{mR}(\boldsymbol{q}_k) = \mathbb{B}_m^T(\boldsymbol{q}_k)\boldsymbol{J}_R(\boldsymbol{q}_k) \in \Re^{\eta_k\times n_k}.$$

Matrices $\mathbb{C}_{cR}(\boldsymbol{q}_k)$ and $\mathbb{C}_{mR}(\boldsymbol{q}_k)$ are always of full column rank, as are $\mathbb{B}_c(\boldsymbol{q}_k)$ and $\mathbb{B}_m(\boldsymbol{q}_k)$. The upper part of (2.73) determines the first-order *differential-motion constraint* of limb e_j, $j \in \{r, l\}$. The two end links in the closed loop move in a synchronous way determined by the two twists $\mathcal{V}_k$. These twists ensure that the closure-link twist is exactly $\mathcal{V}_e$. The lower part of (2.73), on the other hand, determines the instantaneous motion of each end link along the mobility

directions. The end links track thereby the instantaneous motion of the respective contact frame, determined by $\mathcal{V}_k'(\boldsymbol{q}_k)$, along the closure link. These tracking movements of the end links are independent.

Further on, referring to (2.70), the r.h.s. of (2.73) can be expressed as

$$\mathbb{C}_e^T(\boldsymbol{q}_k)\mathcal{V}_e + \mathbb{B}^T(\boldsymbol{q}_k)\mathcal{V}_k' \equiv \overline{\mathcal{V}}_k, \tag{2.76}$$

where

$$\begin{aligned}
\mathbb{B}(\boldsymbol{q}_k) &= \begin{bmatrix}\mathbb{B}_c(\boldsymbol{q}_k) & \mathbb{B}_m(\boldsymbol{q}_k)\end{bmatrix} \in \Re^{6\times 6},\\
\mathbb{C}_e^T(\boldsymbol{q}_k) &= \mathbb{B}^T(\boldsymbol{q}_k)\mathbb{T}_{\overleftarrow{ke}}(\boldsymbol{q}_k),\\
&= \begin{bmatrix}\mathbb{C}_{ce}(\boldsymbol{q}_k) & \mathbb{C}_{me}(\boldsymbol{q}_k)\end{bmatrix}^T,\\
\overline{\mathcal{V}}_k &= \begin{bmatrix}(\overline{\mathcal{V}}_k^c)^T & (\overline{\mathcal{V}}_k^m)^T\end{bmatrix}^T.
\end{aligned}$$

As already shown in (2.66), $\overline{\mathcal{V}}_k^c \perp \overline{\mathcal{V}}_k^m$. Also, the range space of the *constraint* Jacobian of the limb, $\mathcal{J}_{cR}(\boldsymbol{q}_k)$, is an orthogonal complement to that of the limb *mobility* Jacobian, $\mathcal{J}_{mR}(\boldsymbol{q}_k)$. With this notation, (2.73) can be written in compact form as

$$\mathbb{C}_R^T(\boldsymbol{q}_k)\mathcal{V}_R + \mathcal{J}_R(\boldsymbol{q}_k)\dot{\boldsymbol{\theta}}_k = \mathbb{C}_e^T(\boldsymbol{q}_k)\mathcal{V}_e + \mathbb{B}^T(\boldsymbol{q}_k)\mathcal{V}_k', \tag{2.77}$$

where

$$\mathbb{C}_R(\boldsymbol{q}_k) = \begin{bmatrix}\mathbb{C}_{cR}(\boldsymbol{q}_k) & \mathbb{C}_{mR}(\boldsymbol{q}_k)\end{bmatrix} \in \Re^{6\times 6} \tag{2.78}$$

and $\mathcal{J}_R(\boldsymbol{q}_k) = \mathbb{B}^T(\boldsymbol{q}_k)\boldsymbol{J}_R(\boldsymbol{q}_k)$; $\mathbb{B}(\boldsymbol{q}_k)$ plays the role of a permutation matrix that reorders the instantaneous motion equation of the limbs in the closed loop. Matrices $\mathbb{C}_R(\boldsymbol{q}_k)$ and $\mathbb{C}_e(\boldsymbol{q}_k)$ are referred to as the *contact maps* (CMs) of the limb for the root and closure links, respectively. They are composed of CMs in the constrained and mobility directions. Note that for a contact joint with friction, $\mathbb{C}_{(\circ)}(\boldsymbol{q}_k) = \mathbb{C}_{c(\circ)}(\boldsymbol{q}_k)$. On the other hand, the CMs of a completely free "contact joint" are $\mathbb{C}_{(\circ)}(\boldsymbol{q}_k) = \mathbb{C}_{m(\circ)}(\boldsymbol{q}_k)$.

Velocities Within the Closed Chain

The instantaneous-motion equations for the closed loop can be written in the same form as (2.73). We have

$$\begin{bmatrix}\mathbb{C}_{cR}^T(\boldsymbol{q}_e)\\ \mathbb{C}_{mR}^T(\boldsymbol{q}_e)\end{bmatrix}\mathcal{V}_R + \begin{bmatrix}\mathcal{J}_{cR}(\boldsymbol{q}_e)\\ \mathcal{J}_{mR}(\boldsymbol{q}_e)\end{bmatrix}\dot{\boldsymbol{\theta}}_e = \begin{bmatrix}\mathbb{B}_c^T(\boldsymbol{q}_e)\\ \mathbb{B}_m^T(\boldsymbol{q}_e)\end{bmatrix}\begin{bmatrix}\mathcal{V}_{e_r}\\ \mathcal{V}_{e_l}\end{bmatrix}, \tag{2.79}$$

where

$$\begin{aligned}
\mathbb{B}_c^T(\boldsymbol{q}_e) &= \text{diag}\left(\mathbb{B}_c^T(\boldsymbol{q}_{e_r}),\, \mathbb{B}_c^T(\boldsymbol{q}_{e_l})\right) \in \Re^{c_e\times 12},\\
\mathbb{B}_m^T(\boldsymbol{q}_e) &= \text{diag}\left(\mathbb{B}_m^T(\boldsymbol{q}_{e_r}),\, \mathbb{B}_m^T(\boldsymbol{q}_{e_l})\right) \in \Re^{\eta_e\times 12},
\end{aligned}$$

and $\mathcal{J}_{cR}(\boldsymbol{q}_e) \in \Re^{c_e \times n_e}$, $\mathcal{J}_{mR}(\boldsymbol{q}_e) \in \Re^{\eta_e \times n_e}$ denoting the *joint-space constraint* and *end-link mobility* Jacobians of the closed loop, respectively. The parameters are related as follows: $c_e = c_{e_r} + c_{e_l}$, $\eta_e = \eta_{e_r} + \eta_{e_l} = 12 - c_e$. The r.h.s. of (2.79) can be expressed as

$$\begin{bmatrix} \mathbb{C}_{ce}^T(\boldsymbol{q}_e) \\ \mathbb{C}_{me}^T(\boldsymbol{q}_e) \end{bmatrix} \mathcal{V}_e + \mathbb{B}^T(\boldsymbol{q}_e) \begin{bmatrix} \mathcal{V}'_{e_r} \\ \mathcal{V}'_{e_l} \end{bmatrix} \equiv \begin{bmatrix} \overline{\mathcal{V}}_e^c \\ \overline{\mathcal{V}}_e^m \end{bmatrix}, \quad \overline{\mathcal{V}}_e^c \perp \overline{\mathcal{V}}_e^m, \tag{2.80}$$

where

$$\mathbb{B}^T(\boldsymbol{q}_e) = \text{diag}\left(\mathbb{B}_c^T(\boldsymbol{q}_{e_r}), \mathbb{B}_c^T(\boldsymbol{q}_{e_l}), \mathbb{B}_m^T(\boldsymbol{q}_{e_l}), \mathbb{B}_m^T(\boldsymbol{q}_{e_l})\right) \in \Re^{12\times 12}.$$

The CMs of the closed loop are in stacked form

$$\mathbb{C}_{c(\circ)}(\boldsymbol{q}_e) = \left[\mathbb{C}_{c(\circ)}(\boldsymbol{q}_{e_r}) \quad \mathbb{C}_{c(\circ)}(\boldsymbol{q}_{e_l})\right] \in \Re^{6\times c_e}, \tag{2.81}$$

$$\mathbb{C}_{m(\circ)}(\boldsymbol{q}_e) = \left[\mathbb{C}_{m(\circ)}(\boldsymbol{q}_{e_r}) \quad \mathbb{C}_{m(\circ)}(\boldsymbol{q}_{e_l})\right] \in \Re^{6\times \eta_e}. \tag{2.82}$$

With this notation, (2.79) can be written in compact form as

$$\mathbb{C}_R^T(\boldsymbol{q}_e)\mathcal{V}_R + \mathcal{J}_R(\boldsymbol{q}_e)\dot{\boldsymbol{\theta}}_e = \mathbb{C}_e^T(\boldsymbol{q}_e)\mathcal{V}_e + \mathbb{B}^T(\boldsymbol{q}_e)\mathcal{V}'(\boldsymbol{q}_e), \tag{2.83}$$

where

$$\mathbb{C}_{(\circ)}^T(\boldsymbol{q}_e) = \mathbb{B}^T(\boldsymbol{q}_e)\left[\mathbb{T}_{\underleftarrow{e_r(\circ)}}^T(\boldsymbol{q}_{e_r}) \quad \mathbb{T}_{\underleftarrow{e_l(\circ)}}^T(\boldsymbol{q}_{e_l})\right]^T \in \Re^{12\times 6},$$

$$\mathcal{J}_R(\boldsymbol{q}_e) = \mathbb{B}^T(\boldsymbol{q}_e)\boldsymbol{J}_R(\boldsymbol{q}_e) \in \Re^{12\times n_e},$$

$$\mathcal{V}'(\boldsymbol{q}_e) = \left[(\mathcal{V}'_{e_r})^T \quad (\mathcal{V}'_{e_l})^T\right]^T.$$

As an example, consider the frequently occurring case of double stance on stationary ground: $e = F$, $R = B$, $\mathcal{V}_F = \mathbf{0}$. Then, (2.79) becomes

$$\begin{bmatrix} \mathbb{C}_{cR}^T(\boldsymbol{q}_F) \\ \mathbb{C}_{mR}^T(\boldsymbol{q}_F) \end{bmatrix} \mathcal{V}_B + \begin{bmatrix} \mathcal{J}_{cR}(\boldsymbol{q}_F) \\ \mathcal{J}_{mR}(\boldsymbol{q}_F) \end{bmatrix} \dot{\boldsymbol{\theta}}_F = \mathbb{B}^T(\boldsymbol{q}_F) \begin{bmatrix} \mathcal{V}'_{F_r} \\ \mathcal{V}'_{F_l} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \overline{\mathcal{V}}_F^m \end{bmatrix}. \tag{2.84}$$

Note that, since the twist of the loop-closure link is zero ($\mathcal{V}_F = \mathbf{0}$), only relative-motion twists $\mathcal{V}'_{F_j}$ are present. Note also that the projection of these twists along the constrained motion directions results in a zero, under the assumption of *hard constraints*: $\overline{\mathcal{V}}_F^c = \mathbf{0}$. From the upper equation it is apparent that joint rates $\dot{\boldsymbol{\theta}}_F$ ensure the instantaneous motion of the base frame, expressed by twist $\mathcal{V}_B$. From the lower equation, on the other hand, it is apparent that the joint rates also ensure the relative motion of the feet, i.e. sliding on the floor with a twist around the vertical, determined by the composite spatial velocity $\overline{\mathcal{V}}_F^m$. Appropriate choices for the joint rates will be introduced in the following section.

2.10.2 Inverse Kinematics Solution

The inverse kinematics problem for closed loop e can be formulated as follows: "Given the loop-closure and root-link twists $\mathcal{V}_e$ and $\mathcal{V}_R$ and the relative end-link twist $\mathcal{V}'_k$, find the limb joint velocities $\dot{\boldsymbol{\theta}}_k$, $k \in \{e_r,\ e_l\}$." The solution to the above problem can be derived from (2.73). It would be straightforward to solve this equation for the joint velocity that instantaneously satisfies both constraints determined by the r.h.s., i.e. the loop-closure one and the end-link relative velocity one. It is important to note, however, that the former constraint is a physical one while the latter is a task-induced one. Having in mind that physical constraints should always be satisfied, it would be preferable to resort to a solution with a priority structure such that the loop-closure constraints have the higher priority. To this end, first solve the upper part of (2.73) for the limb joint velocity, i.e.

$$\dot{\boldsymbol{\theta}}_k = \boldsymbol{\mathcal{J}}_{cR}^{+}(\boldsymbol{q}_k)\widetilde{\mathcal{V}}_k^c + \left(\boldsymbol{E} - \boldsymbol{\mathcal{J}}_{cR}^{+}(\boldsymbol{q}_k)\boldsymbol{\mathcal{J}}_{cR}(\boldsymbol{q}_k)\right)\dot{\boldsymbol{\theta}}_{ku}, \tag{2.85}$$

where $\widetilde{\mathcal{V}}_k^c = \overline{\mathcal{V}}_k^c - \mathbb{C}_{cR}^T(\boldsymbol{q}_k)\mathcal{V}_R$. The joint velocity $\dot{\boldsymbol{\theta}}_{ku}$ of the unconstrained limb is then determined to satisfy the end-link relative motion task constraint in the lower part of (2.73). As a result, one obtains the following constrained least-squares solution (cf. (2.49)):

$$\begin{aligned}\dot{\boldsymbol{\theta}}_k &= \boldsymbol{\mathcal{J}}_{cR}^{+}(\boldsymbol{q}_k)\widetilde{\mathcal{V}}_k^c + \overline{\boldsymbol{\mathcal{J}}}_{mR}^{+}(\boldsymbol{q}_k)\widetilde{\mathcal{V}}_k^m + \left(\boldsymbol{E} - \boldsymbol{J}_R^{+}(\boldsymbol{q}_k)\boldsymbol{J}_R(\boldsymbol{q}_k)\right)\dot{\boldsymbol{\theta}}_{ku} \\ &= \dot{\boldsymbol{\theta}}_k^{\,c} + \dot{\boldsymbol{\theta}}_k^{\,m} + \dot{\boldsymbol{\theta}}_k^{\,n}, \quad \text{s.t. } \dot{\boldsymbol{\theta}}_k^{\,c} \succ \dot{\boldsymbol{\theta}}_k^{\,m} \succ \dot{\boldsymbol{\theta}}_k^{\,n}.\end{aligned} \tag{2.86}$$

Here $\overline{\boldsymbol{\mathcal{J}}}_{mR}(\boldsymbol{q}_k) = \boldsymbol{\mathcal{J}}_{mR}(\boldsymbol{q}_k)\boldsymbol{N}(\boldsymbol{\mathcal{J}}_{cR}(\boldsymbol{q}_k))$ is the end-link mobility Jacobian restricted by the null space of the limb constraint Jacobian, $\mathcal{N}\left(\boldsymbol{\mathcal{J}}_{cR}(\boldsymbol{q}_k)\right)$. The end-link velocity $\widetilde{\mathcal{V}}_k^m = \overline{\mathcal{V}}_k^m - \boldsymbol{\mathcal{J}}_{mR}(\boldsymbol{q}_k)\boldsymbol{\mathcal{J}}_{cR}^{+}(\boldsymbol{q}_k)\widetilde{\mathcal{V}}_k^c - \mathbb{C}_{mR}^T(\boldsymbol{q}_k)\mathcal{V}_R$. The joint velocity $\dot{\boldsymbol{\theta}}_{ku}$ can be reused to parametrize any remaining DoFs within the null space of the limb Jacobian, $\mathcal{N}\left(\boldsymbol{J}_R(\boldsymbol{q}_k)\right)$. Such DoFs are available when the limb is kinematically redundant ($n_k - 6 > 0$); they determine its self-motion.

On the other hand, when the limb is nonredundant ($n_k = 6$), the last term in the above equation vanishes. This is the case with the leg branches (cf. Fig. 2.13), i.e. $e = F$ and $R = B$. Recall the example in 2.10.1 where a stationary support with *hard contacts* was assumed. Then, $\mathcal{V}_F = \mathbf{0} = \overline{\mathcal{V}}_{F_j}^c$ (cf. (2.84)). The higher-priority component (the first term on the r.h.s. of (2.86)) yields then instantaneous motion in each leg that will only contribute to the base link velocity $\mathcal{V}_B$. The lower-priority component (the second term), on the other hand, will produce a sliding motion within the support plane and/or a rotation around the plane normal. This subtask can be accomplished independently from the respective subtask with the other leg.

Further on, in many cases high-friction contacts at the feet are assumed, s.t. foot velocity $\mathcal{V}_{F_j}$ becomes identically zero (under the assumption of a stationary support surface). This also implies that the constraint basis at the feet $\mathbb{B}_c = \boldsymbol{E}_6$. Hence, $\boldsymbol{\mathcal{J}}_{cB}(\boldsymbol{q}_{F_j})$ becomes identical to the limb Jacobian $\boldsymbol{J}_B(\boldsymbol{q}_{F_j})$ (a square matrix). Since the feet are fixed, the relative twist $\mathcal{V}'_F = \mathbf{0}$. Then, from (2.86) the following unique solution is obtained:

$$\dot{\boldsymbol{\theta}}_{F_j} = \boldsymbol{J}_B^{-1}(\boldsymbol{q}_{F_j})\mathcal{V}_B. \tag{2.87}$$

The closed-loop formulas derived above hold even when a contact breaks and the respective closed loop ceases to exist. For instance, consider the change from a double to a single leg

stance, e.g. when the left foot lifts off the floor. Then, the last equation expresses the inverse kinematics relationship for a nonredundant serial-link limb (the right leg, $j = r$) on a fixed base, provided the contact is maintained.

The limb joint velocities derived from the above equations will be henceforth referred to as *constraint-compatible* or *constraint-consistent*.

2.10.3 Forward Kinematics Solution

The forward kinematics problem for closed loop e can be formulated as follows: "Given an arbitrary loop-root twist $\mathcal{V}_R$ (or loop-closure twist $\mathcal{V}_e$) and a (constraint-consistent) loop joint velocity $\dot{\boldsymbol{\theta}}_e$, find the loop-closure link twist $\mathcal{V}_e$ (or loop-root twist $\mathcal{V}_R$)." This formulation reflects the relative character of the instantaneous motion relation within the closed loop. The solution, obtained from the upper part of (2.79), will be unique only when the rank of each matrix $\mathbb{B}_c^T(\boldsymbol{q}_e),\ \mathbb{C}_{cR}^T(\boldsymbol{q}_e) \in \Re^{c_e \times 12}$, is six. This implies six independent loop constraints, i.e. $c_e = 6$. The closed loop state is described then as *fully constrained*. As an example, consider closed loop H formed by the rod, with cylindrical contact joints (cf. Fig. 2.13). Each contact joint imposes four constraints; there are a total of eight constraints within the closed loop. Note, however, that the constraints are *not independent* and hence the rod is not fully constrained: it can rotate and translate between the hands. To obtain a fully constrained object, it would be sufficient to assume that one of the contacts (e.g. at the right hand) is characterized by high friction. Then, the number of independent constraints in the closed loop will be $c_H = 6$. Assume also that the loop-root (torso) twist $\mathcal{V}_T$ is known. The unique solution for the rod twist is then obtained from the upper part of (2.79) as

$$\mathcal{V}_H = \mathbb{B}_c^{-T}(\boldsymbol{q}_{H_r}) \left(\mathbb{C}_{cR}^T(\boldsymbol{q}_{H_r}) \mathcal{V}_T + \mathcal{J}_{cT}(\boldsymbol{q}_{H_r}) \dot{\boldsymbol{\theta}}_{H_r} \right). \tag{2.88}$$

Matrix $\mathbb{B}_c{}^{-T}(\boldsymbol{q}_{H_r}) \in \Re^{6\times 6}$ represents a spatial transform, regular at all postures. Apparently, the motion of the object is completely determined by the instantaneous motion of the right hand. Arbitrary *constraint-consistent* loop velocities can then be assigned to $\dot{\boldsymbol{\theta}}_H$. The left hand may thereby move relative to the motion of the object, i.e. translating and/or rotating along the unconstrained directions at the left hand contact joint.

In the case $c_e > 6$, matrices $\mathbb{B}_c^T(\boldsymbol{q}_e)$ and $\mathbb{C}_{cR}^T(\boldsymbol{q}_e)$ will have ranks greater than six. The closed loop is then said to be *unilaterally overconstrained*. This situation appears quite frequently in closed loops, $e = F$ (and $R = B$), with a double stance posture. Assuming that the contacts at the feet are characterized by high friction and are *always maintained*, $c_{F_j} = 6$ and $\mathcal{J}_{cB}(\boldsymbol{q}_{F_j}) = \boldsymbol{J}_B(\boldsymbol{q}_{F_j})$. As already clarified above, in this case there is a unique solution to the inverse kinematics problem which would produce the appropriate constraint-consistent joint velocity for the given loop-closure link velocity. This can be verified by inserting the closed-loop joint velocity obtained from (2.87) into the constraint equation (the upper part of (2.79)), under the assumption that $\mathcal{V}_k = \mathbf{0}$. Then we have

$$-\begin{bmatrix} \mathbb{C}_{cB}^T(\boldsymbol{q}_{F_r}) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_{F_l}) \end{bmatrix} \mathcal{V}_B = \begin{bmatrix} \boldsymbol{J}_B(\boldsymbol{q}_{F_r}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_B(\boldsymbol{q}_{F_l}) \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_B^{-1}(\boldsymbol{q}_{F_r}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_B^{-1}(\boldsymbol{q}_{F_l}) \end{bmatrix} \mathcal{V}_B = \begin{bmatrix} \mathcal{V}_B \\ \mathcal{V}_B \end{bmatrix}. \tag{2.89}$$

This implies that $\mathbb{C}_{cB}^T(\boldsymbol{q}_{F_r}) = \mathbb{C}_{cB}^T(\boldsymbol{q}_{F_l}) = -\boldsymbol{E}_6$.

In the case of $c_e < 6$, on the other hand, the rank of matrices ${}_c\mathbb{B}^T(\boldsymbol{q}_e)$ and $\mathbb{C}^T_{cR}(\boldsymbol{q}_e)$ is less than six and the loop is said to be *underconstrained*. An example of an underconstrained H-loop was mentioned above. Another example are frictionless contact joints at the feet, whereby $c_F = 3$. Then, there will be an infinite number of solutions for the loop-root link velocity $\mathcal{V}_B$. The set of solutions can be expressed as a sum of a particular solution, obtained via a generalized inverse, and a homogeneous solution, i.e.

$$\mathcal{V}_B = -\left(\mathbb{C}^T_{cB}(\boldsymbol{q}_F)\right)^{\#} \mathcal{J}_{cB}(\boldsymbol{q}_F)\dot{\boldsymbol{\theta}}_F + \left(\boldsymbol{E} - \mathbb{C}^T_{cB}(\boldsymbol{q}_F)\left(\mathbb{C}^T_{cB}(\boldsymbol{q}_F)\right)^{\#}\right)\mathcal{V}_{Ba}. \tag{2.90}$$

Here $\mathcal{V}_{Ba}$ denotes an arbitrary base link velocity that parametrizes the infinite set. This velocity is projected onto the null space of $\mathbb{C}^T_{cB}(\boldsymbol{q}_F)$ and accounts for the arbitrary velocities of the base resulting from insufficient loop constraints. This means that the motion of the base will not be controllable.

In the last equation, the generalized inverse of a transposed matrix appeared. This type of generalized inverse is frequently used in kinetostatic relations (cf. Section 3.4). It is straightforward to confirm that the identity $(\boldsymbol{X}^T)^{\#} = (\boldsymbol{X}^{\#})^T$ holds, which implies that the order of the transpose/inverse operations is irrelevant. In addition, the notation will be simplified by dropping the brackets. For example, the generalized inverse in the last expression will be denoted as $\mathbb{C}^{\#T}_{cB}(\boldsymbol{q}_F)$.

2.11 DIFFERENTIAL MOTION RELATIONS OF A HUMANOID ROBOT

2.11.1 Quasivelocity, Holonomic and Nonholonomic Contact Constraints

Quasivelocity, holonomic and nonholonomic constraints are related to the integrability of the first- and second-order differential relations. First, recall that *generalized coordinates* are *physical coordinates* that define the DoFs of the system. *Generalized velocity*, on the other hand, may not necessarily be defined as the time derivative of the generalized coordinates. It is possible to define generalized velocity as a linear combination of the time derivatives of the generalized coordinates. For example, as already clarified in Section 2.4.2, the angular velocity components of a body can be represented as linear combinations of the Euler angle derivatives. No physical coordinates are associated with the integrals of the angular velocity components. Thus, the integrals are referred to as the *quasicoordinates* [95]. Furthermore, the term *quasivelocity* can be introduced to distinguish a generalized velocity that incorporates angular velocity components, from generalized velocities that are time differentials of physical coordinates [49].

The differential motion relations will be expressed as functions of the generalized coordinates of the robot; $\boldsymbol{q} = (\mathcal{X}_B,\ \boldsymbol{\theta})$, where $\mathcal{X}_B$ denotes the 6D position of the base link derived with the preferred choice of local coordinates (e.g. Euler angles or Euler parameters; cf. Section 2.4.2) and $\boldsymbol{\theta} = \begin{bmatrix}\boldsymbol{\theta}^T_{F_r} & \boldsymbol{\theta}^T_{F_l} & \boldsymbol{\theta}^T_{H_r} & \boldsymbol{\theta}^T_{H_l}\end{bmatrix}^T \in \Re^n$ stands for the joint angle vector, $n \geq 24$ denoting the total number of joints. Further on, a quasivelocity will be denoted with the usual overdot that signifies differentiation w.r.t. time. Different types of quasivelocities will

be distinguished via a subscript. For example, the quasivelocity associated with the base-link twist $\mathcal{V}_B$, henceforth referred to as the *base quasivelocity*, will be denoted as $\dot{\boldsymbol{q}}_B = (\mathcal{V}_B, \dot{\boldsymbol{\theta}})$. Another type of quasivelocity, quite frequently used in the field of humanoid robotics, will be introduced in Section 2.11.4. The quasivelocity notation enables the omitting of linear transforms like (2.15) throughout the equations, thus rendering them in simpler form. In simulation and control algorithms, though, this type of transform would be required in the numerical integration procedures.

Next, consider the contact constraints. So far, these constraints were assumed holonomic. This is justified in most situations, when the humanoid robot assumes a single or double leg stance posture with high friction, whereby the base link is (unilaterally) fully constrained or overconstrained. In the case of holonomic constraints, the first-order differential motion constraint (the upper part of (2.79)) is integrable. On the other hand, there are specific postures whereby the base link is characterized as underconstrained. For example, when the robot is on hard ground but the feet are rolling, or when the robot is balancing on soft ground. In this case, the contact constraints are said to be nonholonomic. A nonholonomic contact constraint implies that the time integral of the first-order differential motion constraint has no physical meaning. Nonholonomic contact constraints also appear during multifinger object manipulation with rolling contacts without slipping [103]. Note also that nonholonomic constraints do not always stem from contacts: in Chapter 4 the angular momentum conservation constraint will be introduced to account for a mid-air posture of the robot, e.g. during running or jumping. Nonholonomic motion constraints necessitate a respective nonholonomic motion planning approach to arrive at desired states that are not directly accessible via the differential relation.

In the case of holonomic contact constraints, the 6D position of the floating base can be expressed in terms of the joint variables. Indeed, as already clarified in the last section, for a single or double stance posture with high-friction contacts, there is a unique solution to the forward kinematics problem. This implies that the 6D position of the base link can be obtained from the joint angle sensor data. On the other hand, in the case of nonholonomic contact constraints, the 6D position of the base link cannot be obtained from the joint angle data alone; inertial measurement data will be needed to evaluate the state. This is the reason why the 6D position of the base link should be included into the model as a generalized coordinate.

The following discussion will cover both holonomic and nonholonomic constraints.

2.11.2 First-Order Differential Motion Relations Expressed in Terms of Base Quasivelocity

The instantaneous motion relations for independent closed loops derived in Section 2.10.1 apply under the following conditions. First, substitute independent closed-loop generalized coordinates $\boldsymbol{q}_e$ with the generalized coordinates of the robot, $\boldsymbol{q}$ (as defined in 2.11). Next, assume that closed loops are formed by p contact joints. The maximum number of constrained motion directions is then $6p$, i.e. in the case of bilateral contacts. In the general case of mixed unilateral/bilateral constraints, the total number of motion constraints is $c = \sum c_k \leq 6p$. The number of unconstrained motion directions is then determined as $\eta = \sum \eta_k = 6p - c$. For clarity, in the following discussion it will be assumed that contact joints are formed at the end links only ($p = 4$).

Furthermore, the only root link is the base link: $R = B$. The *contact map of the robot w.r.t. the base link* is $\mathbb{C}_B(\boldsymbol{q}) \in \Re^{6\times 24}$, with components $\mathbb{C}_{cB}(\boldsymbol{q}) \in \Re^{6\times c}$ and $\mathbb{C}_{mB}(\boldsymbol{q}) \in \Re^{6\times \eta}$. Jacobian $\boldsymbol{J}_B(\boldsymbol{q}) \in \Re^{24\times n}$ is the *complete Jacobian* of the robot. The twist of the base link, $\mathcal{V}_B$, is determined by the specific task, e.g. to produce a variation in the whole-body posture while maintaining the contacts. The number of closure links, on the other hand, varies with the application task. In the case of a dual-arm manipulation task, two closure links are involved: the floor supporting the feet ($e = F$) and the object ($e = H$). This implies two independent closed loops, as already explained. In the case of a surface cleaning task, on the other hand, the environment (which includes the ground) is the only closure link. Both cases can be handled with the following notation.

The *first-order differential motion constraint* of the humanoid robot can be written in the form of (2.79), i.e.

$$\begin{bmatrix} \mathbb{C}_{cB}^T(\boldsymbol{q}) \\ \mathbb{C}_{mB}^T(\boldsymbol{q}) \end{bmatrix} \mathcal{V}_B + \begin{bmatrix} \mathcal{J}_{cB}(\boldsymbol{q}) \\ \mathcal{J}_{mB}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \overline{\mathcal{V}}^c \\ \overline{\mathcal{V}}^m \end{bmatrix}. \tag{2.91}$$

Note that $\overline{\mathcal{V}}^c \perp \overline{\mathcal{V}}^m$. Jacobians $\mathcal{J}_{cB}(\boldsymbol{q}) \in \Re^{c\times n}$ and $\mathcal{J}_{mB}(\boldsymbol{q}) \in \Re^{\eta\times n}$ will be referred to as the *joint-space constraint* and *mobility* Jacobians of the robot, respectively. The two parts of the above equation have the same inner structure, which is revealed by the following expansion of the upper part:

$$\begin{aligned} \begin{bmatrix} \overline{\mathcal{V}}_{F_r}^c \\ \overline{\mathcal{V}}_{F_l}^c \\ \overline{\mathcal{V}}_{H_r}^c \\ \overline{\mathcal{V}}_{H_r}^c \end{bmatrix} &= \begin{bmatrix} \mathbb{C}_{cB}^T(\boldsymbol{q}_{F_r}) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_{F_l}) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_{H_r}) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_{H_l}) \end{bmatrix} \mathcal{V}_B + \begin{bmatrix} \mathcal{J}_{cB}(\boldsymbol{q}_{F_r}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{J}_{cB}(\boldsymbol{q}_{F_l}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{J}_{cB}(\boldsymbol{q}_{H_r}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{J}_{cB}(\boldsymbol{q}_{H_l}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\theta}}_{F_r} \\ \dot{\boldsymbol{\theta}}_{F_l} \\ \dot{\boldsymbol{\theta}}_{H_r} \\ \dot{\boldsymbol{\theta}}_{H_l} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{B}_c^T(\boldsymbol{q}_{F_r}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{B}_c^T(\boldsymbol{q}_{F_r}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{B}_c^T(\boldsymbol{q}_{H_r}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{B}_c^T(\boldsymbol{q}_{H_l}) \end{bmatrix} \begin{bmatrix} \mathcal{V}_{F_r} \\ \mathcal{V}_{F_l} \\ \mathcal{V}_{H_r} \\ \mathcal{V}_{H_l} \end{bmatrix}. \end{aligned} \tag{2.92}$$

The compact-form notation of (2.91) can be written in analogy to (2.83) as

$$\overline{\mathcal{V}} = \mathbb{C}_B^T(\boldsymbol{q})\mathcal{V}_B + \mathcal{J}_B(\boldsymbol{q})\dot{\boldsymbol{\theta}} = \mathbb{B}^T(\boldsymbol{q})\mathcal{V}(\boldsymbol{q}). \tag{2.93}$$

Here $\mathcal{J}_B(\boldsymbol{q}) = \mathbb{B}^T(\boldsymbol{q})\boldsymbol{J}_B(\boldsymbol{q}) \in \Re^{24\times n}$ denotes the permuted Jacobian of the robot. The vector $\mathcal{V}(\boldsymbol{q})$ collects all contact link twists, i.e.

$$\mathcal{V}(\boldsymbol{q}) = \begin{bmatrix} \mathcal{V}(\boldsymbol{q}_F)^T & \mathcal{V}(\boldsymbol{q}_H)^T \end{bmatrix}^T \in \Re^{24}, \tag{2.94}$$

where $\mathcal{V}(\boldsymbol{q}_e) = \begin{bmatrix} \mathcal{V}_{e_r}^T & \mathcal{V}_{e_l}^T \end{bmatrix}^T \in \Re^{12}$.

In the case of holonomic constraints, the first-order differential relations of the humanoid robot, (2.91), can be written in the following compact form:

$$\begin{bmatrix} \boldsymbol{J}_{cB}(\boldsymbol{q}) \\ \boldsymbol{J}_{mB}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}}_B = \begin{bmatrix} \overline{\mathcal{V}}^c \\ \overline{\mathcal{V}}^m \end{bmatrix}. \tag{2.95}$$

Here $\boldsymbol{J}_{cB} \equiv \left[\mathbb{C}_{cB}^{T} \quad \boldsymbol{\mathcal{J}}_{cB}\right] \in \Re^{c\times(n+6)}$ and $\boldsymbol{J}_{mB} \equiv \left[\mathbb{C}_{mB}^{T} \quad \boldsymbol{\mathcal{J}}_{mB}\right] \in \Re^{\eta\times(n+6)}$ are referred to as the *constraint* and *mobility Jacobian* of the robot, respectively.

Structural Changes

The above notation is general in the sense that *structural changes* can be treated at ease. Consider first the case of two independent closed loops (F and H). In this case, the end-link twists are determined, in accordance with (2.70), as

$$\mathcal{V}_{e_j} = \mathbb{T}_{\overleftarrow{e_j e}}(\boldsymbol{q}_{e_j})\mathcal{V}_e + \mathcal{V}'_{e_j},$$
$$e \in \{F, H\}, \;\; j \in \{r, l\}.$$

The projection of these twists along the constrained motion directions produces the closure-link twist, i.e.

$$\mathcal{V}_e = \mathbb{C}_{ce}^{T}(\boldsymbol{q}_e)\mathcal{V}(\boldsymbol{q}_e).$$

As already noted, the constraint bases annihilate the relative motion twists $\mathcal{V}'_k$.

Next, when one of the closed loops or both cease to exist, one or more of the end links will be completely free. The constraint bases of the respective end links change, s.t. ${}^{k}\mathbb{B}_c = \boldsymbol{0}_6$ and ${}^{k}\mathbb{B}_m = \boldsymbol{E}_6$. Apparently, the notation directly accounts for the single foot stances.

In multicontact tasks, there is a single immobile loop-closure link (e.g. the floor F). Also, in many cases hard contacts can be assumed. Then, the constraint equation of the humanoid robot (the upper part in (2.95)) can be written in the simple form

$$\boldsymbol{J}_{cB}(\boldsymbol{q})\dot{\boldsymbol{q}}_B = \boldsymbol{0}. \tag{2.96}$$

This equation appears quite frequently in the literature.

Constraint-Consistent Joint Velocity

Usually, it is assumed that the number of constraints is less than the DoFs of the robot ($c < n$). Then, the first-order differential motion constraint in (2.91) will be underdetermined. The general solution for the joint velocity can be derived from this equation as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{\mathcal{J}}_{cB}^{+}(\boldsymbol{q})\left(\overline{\mathcal{V}}^{c} - \mathbb{C}_{cB}^{T}(\boldsymbol{q})\mathcal{V}_B\right) + \left(\boldsymbol{E} - \boldsymbol{\mathcal{J}}_{cB}^{+}(\boldsymbol{q})\boldsymbol{\mathcal{J}}_{cB}(\boldsymbol{q})\right)\dot{\boldsymbol{\theta}}_u. \tag{2.97}$$

The two components on the r.h.s. denote the particular and homogeneous solutions. Assuming that all limbs are in nonsingular configuration (such posture is henceforth referred to as *regular*), any desired base-link velocity can be achieved with the help of the particular solution. The homogeneous solution, on the other hand, provides an infinite set of joint-velocity vectors that do not affect the state of the base link whatsoever. The null space of the joint-space constraint Jacobian, $\mathcal{N}(\boldsymbol{\mathcal{J}}_{cB})$, is parametrized by the joint velocity $\dot{\boldsymbol{\theta}}_u$ of the unconstrained robot. This velocity can be determined by additional motion constraints, e.g. from the desired motion of a completely free end link, or the desired motion along the unconstrained directions at the contact joints. To this end, make use of the lower part of (2.91). The resultant

inverse kinematics solution can be written as

$$\begin{aligned}\dot{\boldsymbol{\theta}} &= \mathcal{J}_{cB}^{+}(\boldsymbol{q})\left(\overline{\mathcal{V}}^{c} - \mathbb{C}_{cB}^{T}(\boldsymbol{q})\mathcal{V}_{B}\right) + \overline{\mathcal{J}}_{mB}^{+} + (\boldsymbol{q})\widetilde{\mathcal{V}}^{m} + \left(\boldsymbol{E} - \boldsymbol{J}_{B}^{+}(\boldsymbol{q})\boldsymbol{J}_{B}(\boldsymbol{q})\right)\dot{\boldsymbol{\theta}}_{u} \\ &= \dot{\boldsymbol{\theta}}^{c} + \dot{\boldsymbol{\theta}}^{m} + \dot{\boldsymbol{\theta}}^{n}, \quad \text{s.t. } \dot{\boldsymbol{\theta}}^{c} \succ \dot{\boldsymbol{\theta}}^{m} \succ \dot{\boldsymbol{\theta}}^{n}.\end{aligned} \tag{2.98}$$

Here $\overline{\mathcal{J}}_{mB}(\boldsymbol{q}) = \mathcal{J}_{mB}(\boldsymbol{q})\boldsymbol{N}(\mathcal{J}_{cB})$ is the mobility Jacobian restricted by $\mathcal{N}(\mathcal{J}_{cB}(\boldsymbol{q}))$. The twist $\widetilde{\mathcal{V}}^{m}$ is defined as $\widetilde{\mathcal{V}}^{m} = \overline{\mathcal{V}}^{m} - \mathcal{J}_{m}(\boldsymbol{q})\mathcal{J}_{cB}^{+}(\boldsymbol{q})\left(\overline{\mathcal{V}}^{c} - \mathbb{C}_{cB}^{T}(\boldsymbol{q})\mathcal{V}_{B}\right)$. The joint velocity $\dot{\boldsymbol{\theta}}_{u}$ parametrizes now any remaining DoFs within $\mathcal{N}(\boldsymbol{J}_{B}(\boldsymbol{q}))$ that determine the robot's self-motion. With this hierarchical structure, the highest priority is assigned to the constrained base-link motion, ensured via the joint velocity $\dot{\boldsymbol{\theta}}^{c}$. The second component, $\dot{\boldsymbol{\theta}}^{m}$, ensures the motion of the end-links along the unconstrained motion directions. The lowest-priority component, $\dot{\boldsymbol{\theta}}^{n}$, can be used for additional postural adjustments. Since the highest priority is assigned to the motion constraint, joint velocity (2.98) is referred to as the *constraint-compatible* or *constraint-consistent* joint velocity of the robot.

Constraint-Consistent Generalized Velocity

Regular postures of the robot with single/double stance occur quite frequently; they play an essential role in propulsion and balance. As mentioned in Section 2.11.1, the contact constraints are then characterized as holonomic. Such constraints can be represented by a smooth C^2 vector-valued function, $\boldsymbol{\gamma}(\boldsymbol{q}) = \text{const} \in \Re^{c}$. Furthermore, in many practical tasks hard constraints are assumed. In this case, the first-order differential motion constraint of the robot, (2.96), is valid. This is a homogeneous system of c linear equations in $n+6$ unknowns. In the general case, when the constraints are less than the generalized coordinates ($c < n+6$), the system will be underdetermined. This equation is of the same type as the self-motion equation of a redundant manipulator, (2.28). Hence, the discussion in Section 2.7.1 is relevant. Accordingly, the system constraint matrix $\boldsymbol{J}_{cB}$ will induce an orthogonal decomposition of the generalized coordinate space locally. We have

$$\dot{\boldsymbol{q}}_{u} = \dot{\boldsymbol{q}}_{B} + \dot{\boldsymbol{q}}_{c},\ \dot{\boldsymbol{q}}_{B} \perp \dot{\boldsymbol{q}}_{c}, \tag{2.99}$$

$$\dot{\boldsymbol{q}}_{B} = \boldsymbol{N}(\boldsymbol{J}_{cB})\dot{\boldsymbol{q}}_{u}, \tag{2.100}$$

$$\dot{\boldsymbol{q}}_{c} = (\boldsymbol{E} - \boldsymbol{N}(\boldsymbol{J}_{cB}))\,\dot{\boldsymbol{q}}_{u} = \boldsymbol{J}_{cB}^{+}\boldsymbol{J}_{cB}\dot{\boldsymbol{q}}_{u}. \tag{2.101}$$

Here $\boldsymbol{N}(\boldsymbol{J}_{cB}) = \left(\boldsymbol{E} - \boldsymbol{J}_{cB}^{+}\boldsymbol{J}_{cB}\right)$ is the null space projector. The generalized velocity of the unconstrained system is denoted as $\dot{\boldsymbol{q}}_{u} \in \Re^{n+6}$. Its projection yields generalized velocity $\dot{\boldsymbol{q}}_{B} \in \mathcal{N}(\boldsymbol{J}_{cB})$, a tangential component at the *constraint manifold* (the self-motion manifold in Section 2.7.1). This velocity, henceforth referred to as the *constraint-compatible* or *constraint-consistent* generalized velocity, ensures motion in harmony with the constraints. The generalized velocity $\dot{\boldsymbol{q}}_{c} \in \mathcal{R}^{T}(\boldsymbol{J}_{cB})$, on the other hand, is normal to the constraint manifold. Whenever the constraints are satisfied, $\dot{\boldsymbol{q}}_{c} = \boldsymbol{0}$. It is interesting to note that when the constraints are not satisfied, a nonzero $\dot{\boldsymbol{q}}_{c}$ will ensure that the constraint manifold behaves as an attractor; $\dot{\boldsymbol{q}}_{c}$ will be henceforth be referred to as the *constraint enforcing* generalized velocity.

From (2.99), it is straightforward to derive the respective joint velocity components by employing the following *underactuation filtering* matrix:

$$\boldsymbol{S} = \begin{bmatrix}\boldsymbol{0}_{n\times 6} & \boldsymbol{E}_{n}\end{bmatrix} \in \Re^{n\times(n+6)}. \tag{2.102}$$

With this matrix, the constraint-consistent joint velocity is obtained as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{S}\dot{\boldsymbol{q}}_B = \overline{\boldsymbol{S}}(\boldsymbol{J}_{cB})\dot{\boldsymbol{q}}_u, \tag{2.103}$$

where $\overline{\boldsymbol{S}}(\boldsymbol{J}_{cB}) = \boldsymbol{S}\boldsymbol{N}(\boldsymbol{J}_{cB})$ is the restriction of $\boldsymbol{S}$ by the null space of $\boldsymbol{J}_{cB}$. Similarly, the constraint enforcing joint velocity is obtained as $\dot{\boldsymbol{\theta}}_c = \boldsymbol{S}\dot{\boldsymbol{q}}_c$. The two orthogonal joint velocities, $\dot{\boldsymbol{\theta}}$ and $\dot{\boldsymbol{\theta}}_c$, can be used as *noninterfering* control inputs in kinematics-based motion control; the former will produce the desired constraint-consistent motion along the constraint manifold, the latter can be used to ensure a locally minimized deviation from the manifold during the motion.

Notations based on the system constraint matrix, as above, lead to compactness when used in a general (e.g. QP) solver. In analysis, however, the notation should be used with care since important properties could be easily masked by matrix $\boldsymbol{S}$. Note also that the notation increases the computational burden, which might be critical in some cases (e.g. in the case of real-time control).

2.11.3 Second-Order Differential Motion Constraints and Their Integrability

The differential motion constraints introduced above have to be taken in consideration when deriving the equation of motion of the robot, as will be clarified in Chapter 4. The classical approach is to resort to a second-order representation [12,157]. To this end, differentiate (2.91) w.r.t. time. Then we have

$$\mathbb{C}_{cB}^T(\boldsymbol{q})\dot{\mathcal{V}}_B + \dot{\mathbb{C}}_{cB}^T(\boldsymbol{q})\mathcal{V}_B + \boldsymbol{\mathcal{J}}_{cB}(\boldsymbol{q})\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{\mathcal{J}}}_{cB}(\boldsymbol{q})\dot{\boldsymbol{\theta}} = \dot{\overline{\mathcal{V}}}^c. \tag{2.104}$$

The compact form of this equation (i.e. the time differential of the upper part in (2.95)) is written as

$$\boldsymbol{J}_{cB}(\boldsymbol{q})\ddot{\boldsymbol{q}}_B + \dot{\boldsymbol{J}}_{cB}(\boldsymbol{q})\dot{\boldsymbol{q}}_B = \dot{\overline{\mathcal{V}}}^c. \tag{2.105}$$

Henceforth $\ddot{\boldsymbol{q}}_B = \frac{d}{dt}\dot{\boldsymbol{q}}_B$ will be referred to as the *constraint-consistent generalized acceleration*.

Determining a constraint-consistent generalized acceleration is an important problem. Note that, since the system is kinematically redundant, there is an infinite number of constraint-consistent generalized accelerations. Not all of these accelerations can ensure contact twists $\overline{\mathcal{V}}^c$ that comply with the constraints. This is a problem of integrability as will be shown in what follows. To focus on the problem, assume for the time being hard constraints, s.t. $\dot{\overline{\mathcal{V}}}^c = \boldsymbol{0} = \overline{\mathcal{V}}^c$. The general solution to the above equation is then obtained as

$$\ddot{\boldsymbol{q}}_B = -\boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\dot{\boldsymbol{q}}_B + \boldsymbol{N}\boldsymbol{a}. \tag{2.106}$$

Note that vector $\boldsymbol{a}$ is usually assumed arbitrary. Although mathematically correct, this assumption is problematic, however, from the viewpoint of integrability. The integrability of (2.106) ensures that the respective first-order motion constraint, (2.96), is satisfied during the motion. The lack of integrability, on the other hand, would result in velocity drift, and hence in constraint violation. Such velocity drift has been observed and studied in torque

minimization–based kinematic redundancy resolution schemes. Interested readers are referred to [151,62,88,86,87,123].

To ensure integrability, vector $\boldsymbol{a}$ in (2.106) has to be specified in the following way. First, observe that the constraint-consistent generalized acceleration can also be derived as the time differential of the generalized velocity (2.100), i.e.

$$\ddot{\boldsymbol{q}}_B = \dot{\boldsymbol{N}}\dot{\boldsymbol{q}}_u + \boldsymbol{N}\ddot{\boldsymbol{q}}_u. \tag{2.107}$$

The time derivative of the projection operator is

$$\begin{aligned}\dot{\boldsymbol{N}} &= -\left(\frac{d}{dt}\boldsymbol{J}_{cB}^{+}\right)\boldsymbol{J}_{cB} - \boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB} \\ &= \boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\boldsymbol{J}_{cB}^{+}\boldsymbol{J}_{cB} - \boldsymbol{N}\dot{\boldsymbol{J}}_{cB}^{T}(\boldsymbol{J}_{cB}\boldsymbol{J}_{cB}^{T})^{-1}\boldsymbol{J}_{cB} - \boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB} \\ &= -(\boldsymbol{L}+\boldsymbol{L}^{T}),\end{aligned} \tag{2.108}$$

where $\boldsymbol{L} \equiv \boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\boldsymbol{N}$. In the derivation, the following expression was used[3]:

$$\frac{d}{dt}\boldsymbol{J}_{cB}^{+} = -\boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\boldsymbol{J}_{cB}^{+} + \boldsymbol{N}\dot{\boldsymbol{J}}_{cB}^{T}(\boldsymbol{J}_{cB}\boldsymbol{J}_{cB}^{T})^{-1}. \tag{2.109}$$

Generalized acceleration (2.107) can then be rewritten as

$$\ddot{\boldsymbol{q}}_B = -\boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\boldsymbol{N}\dot{\boldsymbol{q}}_u + \boldsymbol{N}\left(\ddot{\boldsymbol{q}}_u - \dot{\boldsymbol{J}}_{c}^{T}\boldsymbol{J}_{cB}^{+T}\dot{\boldsymbol{q}}_u\right). \tag{2.110}$$

This expression identifies with (2.106) if and only if

$$\boldsymbol{N}\dot{\boldsymbol{q}}_u = \dot{\boldsymbol{q}}_B \tag{2.111}$$

and

$$\left(\ddot{\boldsymbol{q}}_u - \dot{\boldsymbol{J}}_{cB}^{T}\boldsymbol{J}_{cB}^{+T}\dot{\boldsymbol{q}}_u\right) = \boldsymbol{a}. \tag{2.112}$$

Relation (2.111) stands for the constraint-consistent generalized velocity (2.100).

The above result guarantees the integrability of the second-order constraint, in addition to that of the first-order one, via (2.100). Once the initial conditions in the simulation environment are set to satisfy the constraints, i.e. $\boldsymbol{q}(0) \equiv \boldsymbol{q}_0 : \boldsymbol{\gamma}(\boldsymbol{q}_0) = \boldsymbol{0}$ and $\dot{\boldsymbol{q}}(0) \equiv \dot{\boldsymbol{q}}_0 = \boldsymbol{N}\dot{\boldsymbol{q}}_u(0)$, the motion will evolve without constraint violation, provided there are no numerical errors. In practice however, there are difficulties in both aspects, the setting of the initial conditions as well as the presence of numerical errors. Details about the former problem are discussed in [119]. With regard to the latter problem, different approaches have been developed. Baumgarte's method is frequently used [11]; see also [5,155]. Accordingly, correcting terms for the errors in the generalized coordinates and velocities are added to the r.h.s. of the differential

[3] Derived by making use of $\boldsymbol{J}_{cB}^{+} = \boldsymbol{J}_{cB}^{T}(\boldsymbol{J}_{cB}\boldsymbol{J}_{cB}^{T})^{-1}$, $\frac{d}{dt}(\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{E}) \Rightarrow \frac{d}{dt}\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\dot{\boldsymbol{A}}\boldsymbol{A}^{-1}$ for any square and nonsingular $\boldsymbol{A}$, and set $\boldsymbol{A} \equiv \boldsymbol{J}_{cB}\boldsymbol{J}_{cB}^{T}$. The expression appeared first in [117,118].

constraints, in a fashion resembling PD feedback control. This method may not be efficient, though, when the system constraint matrix is ill-conditioned. The errors are amplified then and result in unreasonably large accelerations and respective forces of constraint. Rank deficiency of the system constraint matrix (i.e. the case of dependent constraints), on the other hand, leads to a singularity problem resembling that discussed in Section 2.5.

The constraint-consistent generalized acceleration plays an important role in the dynamic analysis discussed in Chapter 4. It is therefore important to understand the role of each component in (2.110). First, recall the properties pertinent to the inverse differential kinematic relations of an underdetermined system, as explained in Section 2.7 for the case of a kinematically redundant limb. Note that the form of (2.110) is identical to that of (2.40), when the end-link acceleration is assumed zero and the pseudoinverse is used as a generalized inverse. As already mentioned, the generalized coordinate motion subspace is decomposed locally into two orthogonal subspaces, null space $\mathcal{N}(\boldsymbol{J}_{cB})$ and range space $\mathcal{R}(\boldsymbol{J}_{cB}^T)$. The former contains accelerations that are tangent to the constraint manifold at the given posture. There are two tangential components in (2.110), a linear and a nonlinear one. The acceleration $\ddot{\boldsymbol{q}}_u$, appearing in the linear tangential component, can be set in an appropriate way to achieve the desired acceleration without violating the constraints. The nonlinear tangential component can be represented as $-\dot{\boldsymbol{J}}_{cB}^T \boldsymbol{J}_{cB}^{+T} \dot{\boldsymbol{q}}_u = -\dot{\boldsymbol{J}}_{cB}^T (\boldsymbol{J}_{cB} \boldsymbol{J}_{cB}^T)^{-1} \boldsymbol{J}_{cB} \dot{\boldsymbol{q}}_c$, whereby (2.99) and (2.96) were used. This component will be identically zero under constraint-consistent motion ($\dot{\boldsymbol{q}}_c = \boldsymbol{0}$). Conversely, it can be concluded that the component will play an important role for compensating any drift from the constraint manifold and hence for integrability.

There is also a second nonlinear, state-dependent component: $-\boldsymbol{J}_{cB}^{+} \dot{\boldsymbol{J}}_c \dot{\boldsymbol{q}}_B$. This component is normal to the constraint manifold; it accounts for the centripetal/centrifugal accelerations stemming from the nonlinear geometry of the constraints. It can be shown that this component is in fact a constraint enforcing generalized acceleration. To this end, differentiate (2.101) w.r.t. time. Then we have

$$\ddot{\boldsymbol{q}}_c = (\boldsymbol{E} - \boldsymbol{N}) \ddot{\boldsymbol{q}}_u - \dot{\boldsymbol{N}} \dot{\boldsymbol{q}}_u . \tag{2.113}$$

Applying (2.108) for $\dot{\boldsymbol{N}}$, premultiplying first by $\boldsymbol{J}_{cB}$ and then by $\boldsymbol{J}_{cB}^{+}$, and using identities $\boldsymbol{J}_{cB} \boldsymbol{N} = \boldsymbol{0}$ and $\boldsymbol{J}_{cB} \boldsymbol{J}_{cB}^{+} = \boldsymbol{E}$, one arrives at

$$\begin{aligned} \boldsymbol{J}_{cB}^{+} \boldsymbol{J}_{cB} \ddot{\boldsymbol{q}}_c &= \boldsymbol{J}_{cB}^{+} \boldsymbol{J}_{cB} \ddot{\boldsymbol{q}}_u - \boldsymbol{J}_{cB}^{+} \dot{\boldsymbol{J}}_{cB} \boldsymbol{J}_{cB}^{+} \boldsymbol{J}_{cB} \dot{\boldsymbol{q}}_u + \boldsymbol{J}_{cB}^{+} \dot{\boldsymbol{J}}_{cB} \dot{\boldsymbol{q}}_u \\ &= \boldsymbol{J}_{cB}^{+} \boldsymbol{J}_{cB} \ddot{\boldsymbol{q}}_u + \boldsymbol{J}_{cB}^{+} \dot{\boldsymbol{J}}_{cB} \dot{\boldsymbol{q}}_B . \end{aligned} \tag{2.114}$$

Hereby, use was made of $\boldsymbol{J}_{cB}^{+} \boldsymbol{J}_{cB} \dot{\boldsymbol{q}}_u = \dot{\boldsymbol{q}}_c$ (cf. (2.101)) and $\dot{\boldsymbol{q}}_B = \dot{\boldsymbol{q}}_u - \dot{\boldsymbol{q}}_c$. All terms in the above equation are accelerations from $\mathcal{R}(\boldsymbol{J}_{cB}^T)$. Taking the last term on the r.h.s. of (2.114) with minus sign renders its role as a constraint enforcing generalized acceleration.

In the case when motion is confined to the constraint manifold, (2.113) can be rewritten as

$$(\boldsymbol{E} - \boldsymbol{N}) \ddot{\boldsymbol{q}}_B = \dot{\boldsymbol{N}} \dot{\boldsymbol{q}} . \tag{2.115}$$

This relation will be used later on. In this case, all terms in (2.114) are rendered identically zero. On the other hand, when motion drifts apart from the manifold, for any $\ddot{\boldsymbol{q}}_u$, (2.114) will ensure that the constraint manifold behaves as an attractor with regard to such drifting motion.

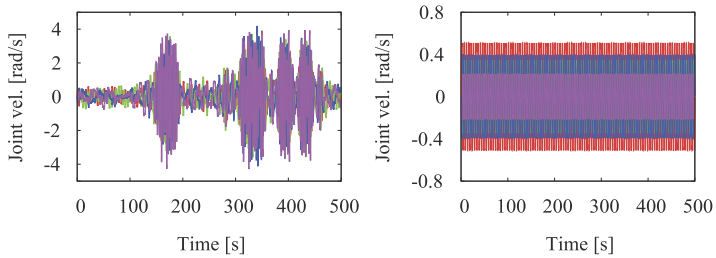

FIGURE 2.14 Acceleration-level redundancy resolution with periodic null-space joint acceleration input. Left (A): the null-space joint velocity obtained via the system-state joint velocity yields aperiodic behavior with large velocity peaks. Right (B): the null-space joint velocity obtained via the integral of the input yields periodic behavior with bounded magnitude.

Example

The goal is to demonstrate the important role of the joint velocity component derived from the null space. To this end, rewrite (2.110) as

$$\ddot{\boldsymbol{q}}_B = -\boldsymbol{J}_{cB}^{+}\dot{\boldsymbol{J}}_{cB}\dot{\boldsymbol{q}}_B + \boldsymbol{N}\left(\ddot{\boldsymbol{q}}_u - \dot{\boldsymbol{J}}_{cB}^{T}\boldsymbol{J}_{cB}^{+T}\dot{\boldsymbol{q}}_\zeta\right). \tag{2.116}$$

This equation will be used in the following two numerical simulations: Case (a) $\dot{\boldsymbol{q}}_\zeta = \dot{\boldsymbol{q}}_B$ and Case (b) $\dot{\boldsymbol{q}}_\zeta = \dot{\boldsymbol{q}}_u$. A fixed-base planar four-link manipulator with fixed end tip is employed (self-motion only, two DoRs). Arbitrary joint acceleration $\ddot{\boldsymbol{q}}_u$ is specified via simple harmonic motion components. Case (a) is the "conventional" approach, whereby the system-state joint velocity $\dot{\boldsymbol{q}}_B$ is employed. The results are shown in Fig. 2.14, in terms of system-state joint velocity. The "conventional" approach (on the l.h.s.) yields aperiodic joint velocity behavior with occasional velocity build-up. The integral approach (Case (b)), on the other hand, yields a periodic joint velocity behavior, as expected. This important result should be considered in control design.

2.11.4 First-Order Differential Motion Relations With Mixed Quasivelocity

The equations derived so far were expressed in terms of base quasivelocity. In the field of humanoid robotics, another type of quasivelocity is also used: $\dot{\boldsymbol{q}}_M = \begin{bmatrix}\mathcal{V}_M^T & \dot{\boldsymbol{\theta}}^T\end{bmatrix}^T$, $\mathcal{V}_M = \begin{bmatrix}\boldsymbol{v}_C^T & \boldsymbol{\omega}_B^T\end{bmatrix}^T$. Vector $\boldsymbol{v}_C$ denotes the velocity of the CoM of the robot, also referred to as the *system centroid*. The position and the instantaneous motion of the CoM play an important role in the static, kinematic, and dynamic relations of the humanoid robot. The two components of $\mathcal{V}_M$ are "mixed" in the sense that the CoM velocity is combined with the angular velocity of the base link; hence the "M" subscripts.

Given the quasivelocity in base coordinates, the CoM velocity can be written as[4]

$$\boldsymbol{v}_C = \boldsymbol{v}_B - [\boldsymbol{r}^{\times}_{\overleftarrow{CB}}]\boldsymbol{\omega}_B + \boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{2.117}$$

[4] This relation is derived in Section 4.6.4.

Here $\boldsymbol{r}_{\overleftarrow{CB}}$ denotes the vector pointing from the base coordinate frame to the CoM. Matrix $\boldsymbol{J}_{\overleftarrow{CB}}$ is the *CoM Jacobian*:

$$\boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{n} M_i \boldsymbol{J}_{vi}(\boldsymbol{\theta}) \in \Re^{3\times n}. \tag{2.118}$$

Here M_i and M denote the mass of Link i and the total mass of the robot, respectively;

$$\boldsymbol{J}_{vi} = \begin{bmatrix} [\boldsymbol{e}_1^{\times}]\boldsymbol{r}_{\overleftarrow{i1}} & [\boldsymbol{e}_2^{\times}]\boldsymbol{r}_{\overleftarrow{i2}} & \cdots & [\boldsymbol{e}_j^{\times}]\boldsymbol{r}_{\overleftarrow{ij}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \in \Re^{3\times n}$$

are Jacobians, $\boldsymbol{e}_j = \boldsymbol{R}_j {}^j\boldsymbol{e}_j$, ${}^j\boldsymbol{e}_j = [0\ 0\ 1]^T$, and $\boldsymbol{r}_{\overleftarrow{ij}}$ is the distance vector from the jth joint axis to the ith link CoM ($1 \le j \le i$).

Closed-chain differential motion relations (2.92) can be rewritten in terms of mixed quasivelocity, as follows. First, decompose the contact map in the constrained motion directions into force and moment component maps, i.e.

$$\mathbb{C}_{cB}(\boldsymbol{q}) \equiv \begin{bmatrix} \mathbb{C}_{cB_f}^T(\boldsymbol{q}) & \mathbb{C}_{cB_m}^T(\boldsymbol{q}) \end{bmatrix}^T \in \Re^{6\times c}. \tag{2.119}$$

Expressions for the two components can be obtained via definition (2.74) of the respective limb contact map, i.e.

$$\begin{aligned} \mathbb{C}_{cB}(\boldsymbol{q}_k) &= \mathbb{T}_{\overleftarrow{kB}}^T(\boldsymbol{q}_k)\mathbb{B}_c(\boldsymbol{q}_k) \\ &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ [\boldsymbol{r}_{\overleftarrow{kB}}^{\times}] & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \mathbb{B}_{c_f}(\boldsymbol{q}_k) & \mathbb{B}_{c_m}(\boldsymbol{q}_k) \end{bmatrix}. \end{aligned}$$

Thus,

$$\begin{aligned} \mathbb{C}_{cB_f}(\boldsymbol{q}_k) &= \mathbb{B}_{c_f}(\boldsymbol{q}_k), \\ \mathbb{C}_{cB_m}(\boldsymbol{q}_k) &= \mathbb{B}_{c_m}(\boldsymbol{q}_k) + [\boldsymbol{r}_{\overleftarrow{kB}}^{\times}]\mathbb{B}_{c_f}(\boldsymbol{q}_k). \end{aligned} \tag{2.120}$$

These components are stacked (as in (2.81)) to obtain an expression for the contact map of the closed chain e in terms of the force $\mathbb{C}_{cB_f}(\boldsymbol{q}_e)$ and moment $\mathbb{C}_{cB_m}(\boldsymbol{q}_e)$ components. These components are in turn stacked to arrive at representation (2.119).

Next, note that via (2.117), the base twist can be expressed in terms of mixed quasivelocity as

$$\mathcal{V}_B = \mathbb{T}_{\overleftarrow{BC}}\mathcal{V}_M - \begin{bmatrix} \boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta}) \\ \boldsymbol{0} \end{bmatrix} \dot{\boldsymbol{\theta}}. \tag{2.121}$$

Using this relation, the first term on the l.h.s. of (2.92) can be represented as

$$\begin{bmatrix} \mathbb{C}_{cB}^T(\boldsymbol{q}_F) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_H) \end{bmatrix} \mathcal{V}_B = \begin{bmatrix} \mathbb{C}_{cC}^T(\boldsymbol{q}_F) \\ \mathbb{C}_{cC}^T(\boldsymbol{q}_H) \end{bmatrix} \mathcal{V}_M - \begin{bmatrix} \mathbb{C}_{cB}^T(\boldsymbol{q}_F) \\ \mathbb{C}_{cB}^T(\boldsymbol{q}_H) \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta}) \\ \boldsymbol{0} \end{bmatrix} \dot{\boldsymbol{\theta}}, \tag{2.122}$$

where $\mathbb{C}_{cC}(\boldsymbol{q}_e)$, $e \in \{F, H\}$ comprise stacked components;

$$\begin{aligned}\mathbb{C}_{cC}(\boldsymbol{q}_k) &= \mathbb{T}^T_{\overleftarrow{BC}}\mathbb{C}_{cB}(\boldsymbol{q}_k) \underset{(2.74)}{=} \mathbb{T}^T_{\overleftarrow{BC}}\mathbb{T}^T_{\overleftarrow{kB}}\mathbb{B}_c(\boldsymbol{q}_k)\\ &= \mathbb{T}^T_{\overleftarrow{kC}}\mathbb{B}_c(\boldsymbol{q}_k) \in \Re^{6\times 6}.\end{aligned} \tag{2.123}$$

With this notation, (2.92) can be rewritten as

$$\begin{aligned}&\begin{bmatrix}\mathbb{C}^T_{cC}(\boldsymbol{q}_F)\\ \mathbb{C}^T_{cC}(\boldsymbol{q}_H)\end{bmatrix}\mathcal{V}_M + \left(\begin{bmatrix}\mathcal{J}_{cB}(\boldsymbol{q}_F) & \mathbf{0}\\ \mathbf{0} & \mathcal{J}_{cB}(\boldsymbol{q}_H)\end{bmatrix} - \begin{bmatrix}\mathbb{C}^T_{cB_f}(\boldsymbol{q}_F)\\ \mathbb{C}^T_{cB_f}(\boldsymbol{q}_H)\end{bmatrix}\boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta})\right)\dot{\boldsymbol{\theta}}\\ &= \begin{bmatrix}\mathbb{B}^T_c(\boldsymbol{q}_F) & \mathbf{0}\\ \mathbf{0} & \mathbb{B}^T_c(\boldsymbol{q}_H)\end{bmatrix}\begin{bmatrix}\mathcal{V}(\boldsymbol{q}_F)\\ \mathcal{V}(\boldsymbol{q}_H)\end{bmatrix}.\end{aligned} \tag{2.124}$$

The general (e.g. nonholonomic constraints) compact-form representation of this equation is

$$\mathbb{C}^T_{cC}(\boldsymbol{q})\mathcal{V}_M + \mathcal{J}_{cM}(\boldsymbol{q})\dot{\boldsymbol{\theta}} = \mathbb{B}^T_c(\boldsymbol{q})\mathcal{V}(\boldsymbol{q}) \equiv \overline{\mathcal{V}}^c, \tag{2.125}$$

where

$$\mathcal{J}_{cM}(\boldsymbol{q}) = \mathcal{J}_{cB}(\boldsymbol{q}) - \mathbb{C}^T_{cB_f}(\boldsymbol{q})\boldsymbol{J}_{\overleftarrow{CB}}(\boldsymbol{\theta}). \tag{2.126}$$

Matrix $\mathbb{C}_{cC}(\boldsymbol{q}) \in \Re^{6\times c}$ is composed of stacked $\mathbb{C}_{cC}(\boldsymbol{q}_e)$ components. This is the contact map of the humanoid robot in the constrained directions, expressed in mixed quasicoordinates; $\mathbb{C}_{cC}$ can be decomposed into force and moment components, i.e.

$$\mathbb{C}_{cC}(\boldsymbol{q}) = \begin{bmatrix}\mathbb{C}^T_{cC_f}(\boldsymbol{q}) & \mathbb{C}^T_{cC_m}(\boldsymbol{q})\end{bmatrix}^T, \tag{2.127}$$

where $\mathbb{C}_{cC_f}(\boldsymbol{q}) = \mathbb{C}_{cB_f}(\boldsymbol{q}) = \mathbb{B}_{c_f}(\boldsymbol{q})$ and $\mathbb{C}_{cC_m}(\boldsymbol{q})$ is composed of stacked components

$$\mathbb{C}_{cC_m}(\boldsymbol{q}_k) = \mathbb{B}_{c_m}(\boldsymbol{q}_k) + [\boldsymbol{r}^{\times}_{\overleftarrow{kC}}]\mathbb{B}_{c_f}(\boldsymbol{q}_k).$$

Velocity relations in the mobility directions can be obtained in a similar fashion, resulting in

$$\mathbb{C}^T_{mC}(\boldsymbol{q})\mathcal{V}_M + \mathcal{J}_{mM}(\boldsymbol{q})\dot{\boldsymbol{\theta}} = \mathbb{B}^T_m(\boldsymbol{q})\mathcal{V}(\boldsymbol{q}) \equiv \overline{\mathcal{V}}^m. \tag{2.128}$$

In the special case of holonomic constraints, (2.125) and (2.128) can be written as

$$\boldsymbol{J}_{cM}(\boldsymbol{q})\dot{\boldsymbol{q}}_M = \overline{\mathcal{V}}^c \tag{2.129}$$

and

$$\boldsymbol{J}_{mM}(\boldsymbol{q})\dot{\boldsymbol{q}}_M = \overline{\mathcal{V}}^m, \tag{2.130}$$

respectively, where $\boldsymbol{J}_{cM}(\boldsymbol{q}) = \begin{bmatrix}\mathbb{C}^T_{cC}(\boldsymbol{q}) & \mathcal{J}_{cM}(\boldsymbol{q})\end{bmatrix}$ and $\boldsymbol{J}_{mM}(\boldsymbol{q}) = \begin{bmatrix}\mathbb{C}^T_{mC}(\boldsymbol{q}) & \mathcal{J}_{mM}(\boldsymbol{q})\end{bmatrix}$.

The complete instantaneous motion relation of the humanoid robot in mixed quasicoordinates can then be obtained by stacking (2.125) and (2.128), i.e.

$$\mathbb{C}_C^T(\boldsymbol{q})\mathcal{V}_M + \mathcal{J}_M(\boldsymbol{q})\dot{\boldsymbol{\theta}} = \mathbb{B}^T(\boldsymbol{q})\mathcal{V}(\boldsymbol{q}) \equiv \begin{bmatrix} \overline{\mathcal{V}}^c \\ \overline{\mathcal{V}}^m \end{bmatrix}. \tag{2.131}$$

Here $\mathbb{C}_C^T(\boldsymbol{q}) = \begin{bmatrix} \mathbb{C}_{cC}(\boldsymbol{q}) & \mathbb{C}_{mC}(\boldsymbol{q}) \end{bmatrix}^T$ and $\mathcal{J}_M(\boldsymbol{q}) = \begin{bmatrix} \mathcal{J}_{cM}^T(\boldsymbol{q}) & \mathcal{J}_{mM}^T(\boldsymbol{q}) \end{bmatrix}^T$ denote the complete contact map and the (permuted) joint-space Jacobian matrix of the robot for mixed quasivelocity, respectively.

Implementation Example

The constraint-consistent joint velocity can be expressed in terms of mixed quasivelocity by making use of (2.125). We have

$$\dot{\boldsymbol{\theta}} = \mathcal{J}_{cM}^+ \left(\overline{\mathcal{V}}^c - \mathbb{C}_{cC}^T \mathcal{V}_M \right) + \boldsymbol{N}(\mathcal{J}_{cM})\, \dot{\boldsymbol{\theta}}_u. \tag{2.132}$$

This equation can be used as a control equation for position-controlled robots, as will be shown in the following example.

Assume a double-stance initial posture on a stationary floor s.t. $\overline{\mathcal{V}}^c = \boldsymbol{0}$. The reference input is determined by the two components of twist $\mathcal{V}_M$: upward CoM translation along a vertical line and base rotation regulated at the initial (zero) value.[5] The resolved joint velocity obtained from the above equation is used to control a small-size robot with parameters similar to those of the HOAP-2 robot [39]. For the numbering of the joints and other relevant data, see Section A.1. The simulation result is shown in Video 2.11-1 [170]. Note that initially, the CoM upward motion is achieved with motion in both the legs and the arms. At a given time instant, the legs become fully stretched. This is a singular configuration w.r.t. the motion of the base link, as explained in Section 2.5. With respect to the desired CoM motion, however, the configuration is not singular. The upward CoM motion continues therefore without destabilization, determined now solely by the upward rotation of the arms. The graphs of the simulation and the snapshots are shown in Fig. 2.15. At around 2.5 s, there is an abrupt change in the motion pattern. The reason is that the legs are fully stretched then and the vertical base motion becomes saturated. Nevertheless, the motion remains stable as apparent from the joint velocity graphs. If the motion were to continue, at some point the arms would become fully stretched upwards. This is a singular posture w.r.t. the desired CoM motion; such posture usually yields destabilization. Another example based on the same approach will be presented in Section 7.5.3.

Further on, the additional arbitrary joint velocity input $\dot{\boldsymbol{\theta}}_u$ can be used to implement other subtasks, such as the end-link motion control along the unconstrained motion directions. The constraint-consistent joint velocity can be written in analogy to (2.98) as

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= \mathcal{J}_{cM}^+(\boldsymbol{q}) \left(\overline{\mathcal{V}}^c - \mathbb{C}_{cC}^T(\boldsymbol{q})\mathcal{V}_M \right) + \overline{\mathcal{J}}_{mM}^+(\boldsymbol{q})\widetilde{\mathcal{V}}^m + \left(\boldsymbol{E} - \boldsymbol{J}_M^+(\boldsymbol{q})\boldsymbol{J}_M(\boldsymbol{q}) \right) \dot{\boldsymbol{\theta}}_u \\ &= \dot{\boldsymbol{\theta}}^c + \dot{\boldsymbol{\theta}}^m + \dot{\boldsymbol{\theta}}^n, \quad \text{s.t. } \dot{\boldsymbol{\theta}}^c \succ \dot{\boldsymbol{\theta}}^m \succ \dot{\boldsymbol{\theta}}^n. \end{aligned} \tag{2.133}$$

[5] The additional arbitrary joint velocity input is not used, i.e. $\dot{\boldsymbol{\theta}}_u = \boldsymbol{0}$.

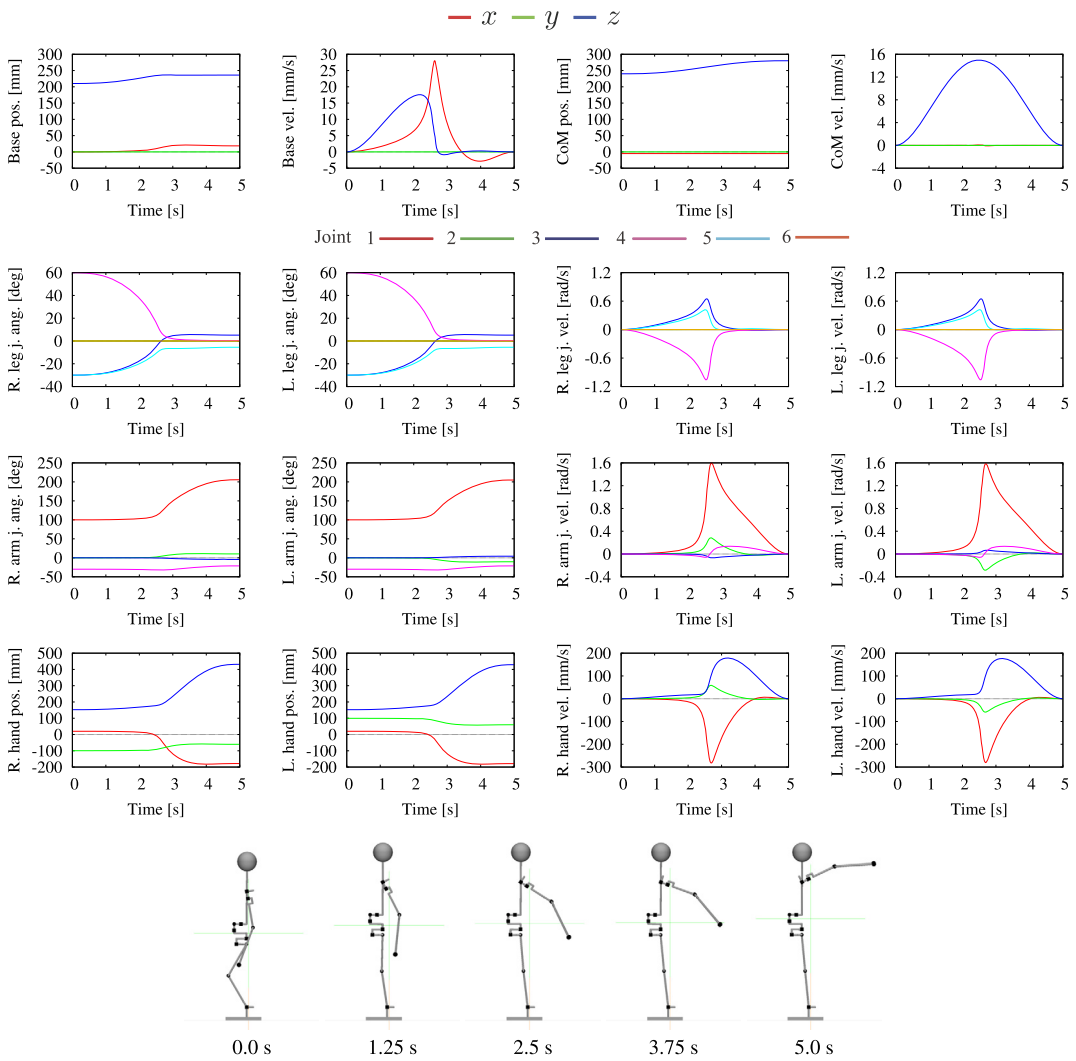

FIGURE 2.15 Simulation of a desired CoM upward motion without hand motion constraints. Initially, the desired motion is achieved with movements in both the legs and the arms. At around 2.5 s, there is an abrupt change in the motion pattern since the legs become fully stretched and the vertical base motion saturates. The CoM motion continues as desired, though, without any disruption.

Here $\overline{\mathcal{J}}_{mM}(\boldsymbol{q}) = \mathcal{J}_{mM}(\boldsymbol{q})\boldsymbol{N}(\mathcal{J}_{cM})$ is the mobility Jacobian restricted by the null space of the constraint Jacobian, $\mathcal{N}(\mathcal{J}_{cM}(\boldsymbol{q}))$. The twist $\widetilde{\mathcal{V}}^m$ is now defined as $\widetilde{\mathcal{V}}^m = \overline{\mathcal{V}}^m - \mathcal{J}_m(\boldsymbol{q})\mathcal{J}_{cM}^{+}(\boldsymbol{q})\left(\overline{\mathcal{V}}^c - \mathbb{C}_{cC}^T(\boldsymbol{q})\mathcal{V}_M\right)$. The joint velocity $\dot{\boldsymbol{\theta}}_u$ parametrizes any remaining DoFs within $\mathcal{N}(\boldsymbol{J}_M(\boldsymbol{q}))$. The above equation will be used for synergy-based motion generation in Section 7.5.

2.11.5 Summary and Discussion

The formulation of differential kinematic relations within two parallel branches forming an independent closed loop can be extended in a straightforward manner to deal with the case of interdependent closed loops formed by multiple parallel branches. The interdependency relation in this case results from the fact that all parallel branches have to accomplish a common main task: ensuring the spatial velocity of the loop-closure link in a consistent way. Thereby, the structure of the inverse kinematics solution can be preserved; only the dimension will increase. Multifinger object manipulation is a good representative example, as already pointed out.

Eq. (2.91) represents the first-order instantaneous motion relations in various types of systems with constraints. In multifinger grasping, the equivalent of the contact map $\mathbb{C}_{cB} : \Re^6 \rightarrow \Re^c$ is called the *grasp map* [104]. It plays an important role in motion/force analysis, grasp planning, and control. It will be shown in what follows that, likewise, $\mathbb{C}_{cB}$ plays an important role in humanoid robotics. The universality of the motion constraint equation can also be confirmed for the case when one or more of the contacts break and an open-loop structure is obtained. Furthermore, it has been shown that with the same equation it is possible to express both holonomic and nonholonomic constraints. With single and double leg stance, the constraints are said to be holonomic. When the base link is underconstrained, a nonholonomic constraint related to the angular momentum conservation may appear. This is also the case when the robot is in mid-air. Angular momentum–related issues will be discussed in Chapter 4.

Finally, it is interesting to note that the first-order differential motion constraint (the upper part of (2.79)) also describes the differential kinematics of parallel-link manipulators [50]. These manipulators comprise a fixed kinematic structure such that the loop-closure link (often referred to as the "traveling plate") connects to the parallel branches via fixed passive joints. The nature of the constraints is similar to that introduced by contact joints. The only difference is that the fixed passive joints cannot brake and thus they cannot induce a change in the kinematic structure, as is the case with humanoid robots and multifinger grasping.

References

[1] A.E. Albert, Regression and the Moore–Penrose Pseudoinverse, 1st edition, Academic Press, 1972.
[2] G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, IEEE Transactions on Robotics 25 (2009) 985–994.
[3] H. Arisumi, S. Miossec, J.-R. Chardonnet, K. Yokoi, Dynamic lifting by whole body motion of humanoid robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 668–675.
[4] U.M. Ascher, H. Chin, S. Reich, Stabilization of DAEs and invariant manifolds, Numerische Mathematik 149 (1994) 131–149.
[5] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
[6] T. Asfour, R. Dillmann, Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, 2003, pp. 1407–1412.
[7] J. Baillieul, Kinematic programming alternatives for redundant manipulators, in: IEEE International Conference on Robotics and Automation, 1985, pp. 722–728.
[8] J. Baillieul, Avoiding obstacles and resolving kinematic redundancy, in: IEEE International Conference on Robotics and Automation, 1986, pp. 1698–1704.

[9] R. Ball, The Theory of Screws: A Study in the Dynamics of a Rigid Body, Hodges, Foster, and Company, 1876.
[10] D. Baraff, Fast contact force computation for nonpenetrating rigid bodies, in: 21st Annual Conference on Computer Graphics and Interactive Techniques – SIGGRAPH '94, ACM Press, New York, New York, USA, 1994, pp. 23–34.
[11] J. Baumgarte, Stabilization of constraints and integrals of motion in dynamical systems, Computer Methods in Applied Mechanics and Engineering 1 (1972) 1–16.
[12] E. Bayo, S. Barbara, A. Avello, Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics, Nonlinear Dynamics 5 (1994) 209–231.
[13] N. Bedrossian, K. Flueckiger, Characterizing spatial redundant manipulator singularities, in: IEEE International Conference on Robotics and Automation, 1991, pp. 714–719.
[14] A. Ben-Israel, T.N. Greville, Generalized Inverses – Theory and Applications, 2nd edition, CMS Books in Mathematics, Springer-Verlag New York, Inc., 2003.
[15] W. Blajer, A geometric unification of constrained system dynamics, Multibody System Dynamics 1 (1997) 3–21.
[16] M. Brandao, L. Jamone, P. Kryczka, N. Endo, K. Hashimoto, A. Takanishi, Reaching for the unreachable: integration of locomotion and whole-body movements for extended visually guided reaching, in: IEEE-RAS International Conference on Humanoid Robots, 2013, pp. 28–33.
[17] O. Brock, O. Khatib, S. Viji, Task-consistent obstacle avoidance and motion behavior for mobile manipulation, in: IEEE International Conference on Robotics and Automation, 2002, pp. 388–393.
[18] J. Burdick, On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds, in: IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 1989, pp. 264–270.
[19] T. Buschmann, S. Lohmeier, H. Ulbrich, Humanoid robot Lola: design and walking control, Journal of Physiology Paris 103 (2009) 141–148.
[20] J.-R. Chardonnet, S. Miossec, A. Kheddar, H. Arisumi, H. Hirukawa, F. Pierrot, K. Yokoi, Dynamic simulator for humanoids using constraint-based method with static friction, in: IEEE International Conference on Robotics and Biomimetics, 2006, pp. 1366–1371.
[21] B. Dariush, G.B. Hammam, D. Orin, Constrained resolved acceleration control for humanoids, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 710–717.
[22] J.K. Davidson, K.H. Hunt, Robots and Screw Theory: Applications of Kinematics and Statics to Robotics, Oxford University Press, 2004.
[23] M. De Lasa, A. Hertzmann, Prioritized optimization for task-space control, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 5755–5762.
[24] A. De Luca, G. Oriolo, Modelling and control of nonholonomic mechanical systems, in: J. Angeles, A. Kecskeméthy (Eds.), Kinematics and Dynamics of Multi-Body Systems, in: CISM International Centre for Mechanical Sciences, vol. 360, Springer, Vienna, 1995, pp. 277–342.
[25] A. Del Prete, F. Romano, L. Natale, G. Metta, G. Sandini, F. Nori, Prioritized optimal control, in: IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014, pp. 2540–2545.
[26] J. Denavit, R. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, in: Trans. of the ASME, Journal of Applied Mechanics 22 (1955) 215–221.
[27] A. Dietrich, C. Ott, A. Albu-Schaffer, Multi-objective compliance control of redundant manipulators: hierarchy, control, and stability, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3043–3050.
[28] A. Dietrich, T. Wimbock, A. Albu-Schaffer, G. Hirzinger, Integration of reactive, torque-based self-collision avoidance into a task hierarchy, IEEE Transactions on Robotics 28 (2012) 1278–1293.
[29] A. Dietrich, T. Wimbock, A. Albu-Schaffer, G. Hirzinger, Reactive whole-body control: dynamic mobile manipulation using a large number of actuated degrees of freedom, IEEE Robotics & Automation Magazine 19 (2012) 20–33.
[30] K.L. Doty, C. Melchiorri, C. Bonivento, A theory of generalized inverses applied to robotics, The International Journal of Robotics Research 12 (1993) 1–19.
[31] J. Duffy, The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces, Journal of Robotic Systems 7 (1990) 139–144.
[32] R. Ellis, S. Ricker, Two numerical issues in simulating constrained robot dynamics, IEEE Transactions on Systems, Man and Cybernetics 24 (1994) 19–27.
[33] A. Escande, N. Mansard, P.-B. Wieber, Fast resolution of hierarchized inverse kinematics with inequality constraints, in: IEEE International Conference on Robotics and Automation, 2010, pp. 3733–3738.

[34] A. Escande, N. Mansard, P.-B. Wieber, Hierarchical quadratic programming: fast online humanoid-robot motion generation, The International Journal of Robotics Research 33 (2014) 1006–1028.
[35] R. Featherstone, Rigid Body Dynamics Algorithms, Springer Science+Business Media, LLC, Boston, MA, 2008.
[36] R. Featherstone, Exploiting sparsity in operational-space dynamics, The International Journal of Robotics Research 29 (2010) 1353–1368.
[37] R. Featherstone, A. Fijany, A technique for analyzing constrained rigid-body systems, and its application to the constraint force algorithm, IEEE Transactions on Robotics and Automation 15 (1999) 1140–1144.
[38] S. Feng, E. Whitman, X. Xinjilefu, C.G. Atkeson, Optimization-based full body control for the DARPA robotics challenge, Journal of Field Robotics 32 (2015) 293–312.
[39] Fujitsu, Miniature Humanoid Robot HOAP-2 Manual, 1st edition, Fujitsu Automation Co., Ltd, 2004 (in Japanese).
[40] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, H. Hirukawa, UKEMI: falling motion control to minimize damage to biped humanoid robot, in: IEEE/RSJ International Conference on Intelligent Robots and System, 2002, pp. 2521–2526.
[41] K. Glass, R. Colbaugh, D. Lim, H. Seraji, Real-time collision avoidance for redundant manipulators, IEEE Transactions on Robotics and Automation 11 (1995) 448–457.
[42] G.H. Golub, C.F. Van Loan, Matrix Computations, Johns Hopkins University Press, 1996.
[43] A. Goswami, S.-k. Yun, U. Nagarajan, S.-H. Lee, K. Yin, S. Kalyanakrishnan, Direction-changing fall control of humanoid robots: theory and experiments, Autonomous Robots 36 (2013) 199–223.
[44] G.B. Hammam, P.M. Wensing, B. Dariush, D.E. Orin, Kinodynamically consistent motion retargeting for humanoids, International Journal of Humanoid Robotics 12 (2015) 1550017.
[45] H. Hanafusa, T. Yoshikawa, Y. Nakamura, Analysis and control of articulated robot arms with redundancy, in: Prep. of the IFAC '81 World Congress, 1981, pp. 78–83.
[46] N. Handharu, J. Yoon, G. Kim, Gait pattern generation with knee stretch motion for biped robot using toe and heel joints, in: IEEE-RAS International Conference on Humanoid Robots, Daejeon, Korea, 2008, pp. 265–270.
[47] Y. Harada, J. Takahashi, D. Nenchev, D. Sato, Limit cycle based walk of a powered 7DOF 3D biped with flat feet, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 3623–3628.
[48] S. Hayati, Hybrid position/force control of multi-arm cooperating robots, in: IEEE International Conference on Robotics and Automation, 1986, pp. 82–89.
[49] P. Herman, K. Kozłowski, A survey of equations of motion in terms of inertial quasi-velocities for serial manipulators, Archive of Applied Mechanics 76 (2006) 579–614.
[50] K.H. Hunt, Structural kinematics of in-parallel-actuated robot-arms, Journal of Mechanisms, Transmissions, and Automation in Design 105 (1983) 705–712.
[51] A. Ijspeert, J. Nakanishi, S. Schaal, Movement imitation with nonlinear dynamical systems in humanoid robots, in: IEEE International Conference on Robotics and Automation, 2002, pp. 1398–1403.
[52] H. Isermann, Linear lexicographic optimization, OR Spektrum 4 (1982) 223–228.
[53] S. Kajita, H. Hirukawa, K. Harada, K. Yokoi, Introduction to Humanoid Robotics, Springer Verlag, Berlin, Heidelberg, 2014.
[54] S. Kajita, T. Nagasaki, K. Kaneko, H. Hirukawa, ZMP-based biped running control, IEEE Robotics & Automation Magazine 14 (2007) 63–72.
[55] K. Kameta, A. Sekiguchi, Y. Tsumaki, Y. Kanamiya, Walking control around singularity using a spherical inverted pendulum with an underfloor pivot, in: IEEE-RAS International Conference on Humanoid Robots, 2007, pp. 210–215.
[56] K. Kameta, A. Sekiguchi, Y. Tsumaki, D. Nenchev, Walking control using the SC approach for humanoid robot, in: IEEE-RAS International Conference on Humanoid Robots, 2005, pp. 289–294.
[57] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, N. Kanehira, Humanoid robot HRP-4 – humanoid robotics platform with lightweight and slim body, in: IEEE International Conference on Intelligent Robots and Systems, 2011, pp. 4400–4407.
[58] O. Kanoun, Real-time prioritized kinematic control under inequality constraints for redundant manipulators, in: H. Durrant-Whyte, N. Roy, P. Abbeel (Eds.), Robotics: Science and Systems VII, MIT Press, 2012, pp. 145–152.
[59] O. Kanoun, F. Lamiraux, P.-B. Wieber, Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task, IEEE Transactions on Robotics 27 (2011) 785–792.
[60] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, J.-P. Laumond, Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots, in: IEEE International Conference on Robotics and Automation, 2009, pp. 2939–2944.

[61] V.N. Katsikis, D. Pappas, The restricted weighted generalized inverse of a matrix, The Electronic Journal of Linear Algebra 22 (2011) 1156–1167.
[62] K. Kazerounian, Z. Wang, Global versus local optimization in redundancy resolution of robotic manipulators, The International Journal of Robotics Research 7 (1988) 3–12.
[63] F. Keith, P.-B.B. Wieber, N. Mansard, A. Kheddar, Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations, in: IEEE International Conference on Intelligent Robots and Systems, 2011, pp. 3887–3892.
[64] J. Kerr, B. Roth, Analysis of multifingered hands, The International Journal of Robotics Research 4 (1986) 3–17.
[65] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research 5 (1986) 90–98.
[66] O. Khatib, L. Sentis, J. Park, J. Warren, Whole-body dynamic behavior and control of human-like robots, International Journal of Humanoid Robotics 01 (2004) 29–43.
[67] C. Klein, K.-B. Kee, The nature of drift in pseudoinverse control of kinematically redundant manipulators, IEEE Transactions on Robotics and Automation 5 (1989) 231–234.
[68] C.A. Klein, B.E. Blaho, Dexterity measures for the design and control of kinematically redundant manipulators, The International Journal of Robotics Research 6 (1987) 72–83.
[69] M.S. Konstantinov, M.D. Markov, D.N. Nenchev, Kinematic control of redundant manipulators, in: 11th International Symposium on Industrial Robots, Tokyo, Japan, 1981, pp. 561–568.
[70] S. Kotosaka, H. Ohtaki, Selective utilization of actuator for a humanoid robot by singular configuration, Journal of the Robotics Society of Japan 25 (2007) 115–121 (in Japanese).
[71] K. Kreutz-Delgado, M. Long, H. Seraji, Kinematic analysis of 7-DOF manipulators, The International Journal of Robotics Research 11 (1992) 469–481.
[72] R. Kurazume, S. Tanaka, M. Yamashita, T. Hasegawa, K. Yoneda, Straight legged walking of a biped robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 337–343.
[73] H.G. Kwatny, G. Blankenship, Nonlinear Control and Analytical Mechanics: A Computational Approach, Springer-Verlag, Birkhäuser, New York, 2000.
[74] A. Laulusa, O.A. Bauchau, Review of classical approaches for constraint enforcement in multibody systems, Journal of Computational and Nonlinear Dynamics 3 (2008) 011004.
[75] J. Lee, N. Mansard, J. Park, Intermediate desired value approach for task transition of robots in kinematic control, IEEE Transactions on Robotics 28 (2012) 1260–1277.
[76] S.-H. Lee, A. Goswami, Fall on backpack: damage minimizing humanoid fall on targeted body segment using momentum control, in: ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2011, pp. 703–712.
[77] S. Lengagne, J. Vaillant, E. Yoshida, A. Kheddar, Generation of whole-body optimal dynamic multi-contact motions, The International Journal of Robotics Research 32 (2013) 1104–1119.
[78] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Quarterly of Applied Mathematics II (1944) 164–168.
[79] Z. Li, N.G. Tsagarikis, D.G. Caldwell, B. Vanderborght, Trajectory generation of straightened knee walking for humanoid robot iCub, in: IEEE International Conference on Control, Automation, Robotics and Vision, 2010, pp. 2355–2360.
[80] A. Liegeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Transactions on Systems, Man and Cybernetics 7 (1977) 868–871.
[81] M. Liu, A. Micaelli, P. Evrard, A. Escande, C. Andriot, Interactive virtual humans: a two-level prioritized control framework with wrench bounds, IEEE Transactions on Robotics 28 (2012) 1309–1322.
[82] M. Liu, Y. Tan, V. Padois, Generalized hierarchical control, Autonomous Robots 40 (2016) 17–31.
[83] D.G. Luenberger, Y. Ye, Linear and Nonlinear Programming, 4th edition, Springer International Publishing, 2016.
[84] R.C. Luo, T.-W. Lin, Y.-H. Tsai, Analytical inverse kinematic solution for modularized 7-DoF redundant manipulators with offsets at shoulder and wrist, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, Il, USA, 2014, pp. 516–521.
[85] S. Luo, S. Ahmad, Predicting the drift motion for kinematically redundant robots, IEEE Transactions on Systems, Man and Cybernetics 22 (1992) 717–728.
[86] S. Ma, S. Hirose, D.N. Nenchev, Improving local torque optimization techniques for redundant robotic mechanisms, Journal of Robotic Systems 8 (1991) 75–91.

[87] S. Ma, D.N. Nenchev, Local torque minimization for redundant manipulators: a correct formulation, Robotica 14 (1996) 235–239.
[88] A. Maciejewski, Kinetic limitations on the use of redundancy in robotic manipulators, IEEE Transactions on Robotics and Automation 7 (1991) 205–210.
[89] A.A. Maciejewski, C.A. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, The International Journal of Robotics Research 4 (1985) 109–117.
[90] A.A. Maciejewski, C.A. Klein, The singular value decomposition: computation and applications to robotics, The International Journal of Robotics Research 8 (1989) 63–79.
[91] N. Mansard, F. Chaumette, Task sequencing for high-level sensor-based control, IEEE Transactions on Robotics 23 (2007) 60–72.
[92] N. Mansard, O. Khatib, Continuous control law from unilateral constraints, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3359–3364.
[93] N. Mansard, O. Khatib, A. Kheddar, A unified approach to integrate unilateral constraints in the stack of tasks, IEEE Transactions on Robotics 25 (2009) 670–685.
[94] N. Mansard, A. Remazeilles, F. Chaumette, Continuity of varying-feature-set control laws, IEEE Transactions on Automatic Control 54 (2009) 2493–2505.
[95] L. Meirovitch, Methods of Analytical Dynamics, Dover Publications, Inc., Mineola, New York, 2012.
[96] P. Meseguer, N. Bouhmala, T. Bouzoubaa, M. Irgens, M. Sánchez, Current approaches for solving over-constrained problems, Constraints 8 (2003) 9–39.
[97] N. Minamide, K. Nakamura, A restricted pseudoinverse and its application to constrained minima, SIAM Journal on Applied Mathematics 19 (1970) 167–177.
[98] S. Miyahara, D.N. Nenchev, Singularity-consistent approach/departure to/from a singular posture, Robotic Life Support Laboratory, Tokyo City University, 2017 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00009-2.
[99] S. Miyahara, D.N. Nenchev, Singularity-consistent motion "through" a singular posture, Robotic Life Support Laboratory, Tokyo City University, 2017 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00009-2.
[100] D. Montana, The kinematics of contact and grasp, The International Journal of Robotics Research 7 (1988) 17–32.
[101] M. Morisawa, S. Kajita, K. Kaneko, K. Harada, F. Kanehiro, K. Fujiwara, H. Hirukawa, Pattern generation of biped walking constrained on parametric surface, in: IEEE International Conference on Robotics and Automation, 2005, pp. 2405–2410.
[102] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot, in: IEEE International Conference on Robotics and Automation, 2015, pp. 5682–5689.
[103] R.M. Murray, Nonlinear control of mechanical systems: a Lagrangian perspective, Annual Reviews in Control 21 (1997) 31–42.
[104] R.M. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.
[105] Y. Nakamura, Advanced Robotics: Redundancy and Optimization, Addison–Wesley Publishing Company, 1991.
[106] Y. Nakamura, H. Hanafusa, Inverse kinematic solutions with singularity robustness for robot manipulator control, Journal of Dynamic Systems, Measurement, and Control 108 (1986) 163.
[107] Y. Nakamura, K. Yamane, Dynamics computation of structure-varying kinematic chains and its application to human figures, IEEE Transactions on Robotics and Automation 16 (2000) 124–134.
[108] S. Nakaoka, S. Hattori, F. Kanehiro, S. Kajita, H. Hirukawa, Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 2007, pp. 3641–3647.
[109] D.N. Nenchev, Redundancy resolution through local optimization: a review, Journal of Robotic Systems 6 (1989) 769–798.
[110] D.N. Nenchev, Restricted Jacobian matrices of redundant manipulators in constrained motion tasks, The International Journal of Robotics Research 11 (1992) 584–597.
[111] D.N. Nenchev, Tracking manipulator trajectories with ordinary singularities: a null space-based approach, The International Journal of Robotics Research 14 (1995) 399–404.
[112] D.N. Nenchev, Z.M. Sotirov, Dynamic task-priority allocation for kinematically redundant robotic mechanisms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Munich, Germany, 1994, pp. 518–524.

[113] D.N. Nenchev, Y. Tsumaki, M. Uchiyama, Singularity-consistent behavior of telerobots: theory and experiments, The International Journal of Robotics Research 17 (1998) 138–152.
[114] D.N. Nenchev, Y. Tsumaki, M. Uchiyama, Singularity-consistent parameterization of robot motion and control, The International Journal of Robotics Research 19 (2000) 159–182.
[115] D.N. Nenchev, Recursive local kinematic inversion with dynamic task-priority allocation, in: IEEE International Conference on Robotics and Automation, IEEE Comput. Soc. Press, Munich, Germany, 1994, pp. 2698–2703.
[116] D.N. Nenchev, Y. Tsumaki, M. Uchiyama, Real-time motion control in the neighborhood of singularities: a comparative study between the SC and the DLS methods, in: IEEE International Conference on Robotics and Automation, 1999, pp. 506–511.
[117] G. Niemeyer, J.-J. Slotine, Computational algorithms for adaptive compliant motion, in: International Conference on Robotics and Automation, 1989, pp. 566–571.
[118] G. Niemeyer, J.-J.E. Slotine, Adaptive Cartesian control of redundant manipulators, in: American Control Conference, 1990, pp. 234–241.
[119] P.E. Nikravesh, Initial condition correction in multibody dynamics, Multibody System Dynamics 18 (2007) 107–115.
[120] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H.-O. Lim, A. Takanishi, Development of a new humanoid robot WABIAN-2, in: IEEE International Conference on Robotics and Automation, 2006, pp. 76–81.
[121] Y. Ogura, H.-O. Lim, A. Takanishi, Stretch walking pattern generation for a biped humanoid robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 352–357.
[122] Y. Oh, W.K. Chung, Y. Youm, l.H. Suh, A passivity-based motion control of redundant manipulators using weighted decomposition of joint space, in: 8th International Conference on Advanced Robotics, 1997, pp. 125–131.
[123] K. O'Neil, Divergence of linear acceleration-based redundancy resolution schemes, IEEE Transactions on Robotics and Automation 18 (2002) 625–631.
[124] C. Ott, A. Dietrich, A. Albu-Schäffer, A. Albu-Schaffer, Prioritized multi-task compliance control of redundant manipulators, Automatica 53 (2015) 416–423.
[125] D. Pai, U.M. Ascher, P. Kry, Forward dynamics algorithms for multibody chains and contact, in: IEEE International Conference on Robotics and Automation, 2000, pp. 857–863.
[126] H.A. Park, M.A. Ali, C. Lee, Closed-form inverse kinematic position solution for humanoid robots, International Journal of Humanoid Robotics 09 (2012) 1250022.
[127] I.-W. Park, J.-Y. Kim, J. Lee, J.-H. Oh, Mechanical design of the humanoid robot platform HUBO, Advanced Robotics 21 (2007) 1305–1322.
[128] J. Park, Control Strategies for Robots in Contact, Ph.D. thesis, Stanford University, USA, 2006.
[129] J. Park, W. Chung, Y. Youm, On dynamical decoupling of kinematically redundant manipulators, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999, pp. 1495–1500.
[130] T. Petrič, A. Gams, J. Babič, L. Žlajpah, Reflexive stability control framework for humanoid robots, Autonomous Robots 34 (2013) 347–361.
[131] T. Petrič, L. Žlajpah, Smooth continuous transition between tasks on a kinematic control level: obstacle avoidance as a control problem, Robotics and Autonomous Systems 61 (2013) 948–959.
[132] M.H. Raibert, J.J. Craig, Hybrid position/force control of manipulators, Journal of Dynamic Systems, Measurement, and Control 103 (1981) 126–133.
[133] A. Rennuit, A. Micaelli, X. Merlhiot, C. Andriot, F. Guillaume, N. Chevassus, D. Chablat, P. Chedmail, Passive control architecture for virtual humans, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 1432–1437.
[134] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, J.Y. Fourquet, Dynamic whole-body motion generation under rigid contacts and other unilateral constraints, IEEE Transactions on Robotics 29 (2013) 346–362.
[135] H. Sadeghian, L. Villani, M. Keshmiri, B. Siciliano, Dynamic multi-priority control in redundant robotic systems, Robotica 31 (2013) 1–13.
[136] L. Sciavicco, B. Siciliano, Modelling and Control of Robot Manipulators, Springer Science & Business Media, 2000.
[137] A. Sekiguchi, Y. Atobe, K. Kameta, D. Nenchev, Y. Tsumaki, On motion generation for humanoid robot by the SC approach, in: Annual Conference of the Robotics Society of Japan, 2003 (in Japanese), p. 2A27.
[138] L. Sentis, O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, International Journal of Humanoid Robotics 02 (2005) 505–518.

[139] L. Sentis, O. Khatib, Compliant control of multicontact and center-of-mass behaviors in humanoid robots, IEEE Transactions on Robotics 26 (2010) 483–501.
[140] L. Sentis, J. Petersen, R. Philippsen, Implementation and stability analysis of prioritized whole-body compliant controllers on a wheeled humanoid robot in uneven terrains, Autonomous Robots 35 (2013) 301–319.
[141] H. Seraji, Configuration control of redundant manipulators: theory and implementation, IEEE Transactions on Robotics and Automation 5 (1989) 472–490.
[142] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, K. Kosuge, Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution, IEEE Transactions on Robotics 24 (2008) 1131–1142.
[143] B. Siciliano, Kinematic control of redundant robot manipulators: a tutorial, Journal of Intelligent & Robotic Systems 3 (1990) 201–212.
[144] B. Siciliano, J.-J. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, in: Fifth International Conference on Advanced Robotics, 1991, pp. 1211–1216.
[145] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, A. Kheddar, Real-time (self)-collision avoidance task on a HRP-2 humanoid robot, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3200–3205.
[146] D.E. Stewart, J.C. Trinkle, An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction, International Journal for Numerical Methods in Engineering 39 (1996) 2673–2691.
[147] G. Strang, Linear Algebra and Its Applications, 4th edition, Cengage Learning, July 19, 2005.
[148] J. Stuelpnagel, On the parametrization of the three-dimensional rotation group, SIAM Review 6 (1964) 422–430.
[149] T. Sugihara, Solvability-unconcerned inverse kinematics by the Levenberg–Marquardt method, IEEE Transactions on Robotics 27 (2011) 984–991.
[150] H. Sugiura, M. Gienger, H. Janssen, C. Goerick, Real-time collision avoidance with whole body motion control for humanoid robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2053–2058.
[151] K. Suh, J. Hollerbach, Local versus global torque optimization of redundant manipulators, in: IEEE International Conference on Robotics and Automation, 1987, pp. 619–624.
[152] K. Takahashi, M. Noda, D. Nenchev, Y. Tsumaki, A. Sekiguchi, Static walk of a humanoid robot based the singularity-consistent method, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 5484–5489.
[153] S. Taki, D.N. Nenchev, A novel singularity-consistent inverse kinematics decomposition for S-R-S type manipulators, in: IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 5070–5075.
[154] J.C. Trinkle, J.-S. Pang, S. Sudarsky, G. Lo, On dynamic multi-rigid-body contact problems with Coulomb friction, ZAMM – Zeitschrift für Angewandte Mathematik und Mechanik (Journal of Applied Mathematics and Mechanics) 77 (1997) 267–279.
[155] F.-C. Tseng, Z.-D. Ma, G.M. Hulbert, Efficient numerical solution of constrained multibody dynamics systems, Computer Methods in Applied Mechanics and Engineering 192 (2003) 439–472.
[156] Y. Tsumaki, D. Nenchev, S. Kotera, M. Uchiyama, Teleoperation based on the adjoint Jacobian approach, IEEE Control Systems Magazine 17 (1997) 53–62.
[157] F. Udwadia, R.E. Kalaba, What is the general form of the explicit equations of motion for constrained mechanical systems?, Journal of Applied Mechanics 69 (2002) 335.
[158] C.W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods, IEEE Transactions on Systems, Man and Cybernetics 16 (1986) 93–101.
[159] D. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Transactions on Man-Machine Systems 10 (1969) 47–53.
[160] D. Whitney, Force feedback control of manipulator fine motions, Journal of Dynamic Systems, Measurement, and Control 99 (1977) 91–97.
[161] T. Wimbock, D. Nenchev, A. Albu-Schaffer, G. Hirzinger, Experimental study on dynamic reactionless motions with DLR's humanoid robot Justin, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, 2009, pp. 5481–5486.
[162] K. Yamane, Y. Nakamura, Dynamics filter – concept and implementation of online motion generator for human figures, IEEE Transactions on Robotics and Automation 19 (2003) 421–432.

[163] K. Yamane, Y. Nakamura, A numerically robust LCP solver for simulating articulated rigid bodies in contact, in: Robotics: Science and Systems IV, MIT Press, Zurich, Switzerland, 2008, pp. 89–104.
[164] Y. Yokokohji, T. Toyoshima, T. Yoshikawa, Efficient computational algorithms for trajectory control of free-flying space robots with multiple arms, IEEE Transactions on Robotics and Automation 9 (1993) 571–580.
[165] S. Yoon, R.M. Howe, D.T. Greenwood, Geometric elimination of constraint violations in numerical simulation of Lagrangian equations, Journal of Mechanical Design 116 (1994) 1058–1064.
[166] T. Yoshikawa, Manipulability of robotic mechanisms, The International Journal of Robotics Research 4 (1985) 3–9.
[167] T. Yoshikawa, Dynamic hybrid position/force control of robot manipulators–description of hand constraints and calculation of joint driving force, IEEE Journal on Robotics and Automation 3 (1987) 386–392.
[168] T. Yoshikawa, Analysis and control of robot arms with redundancy, in: First International Symposium on Robotics Research, Pittsburg, Pennsylvania, MIT Press, Cambridge, MA, 1994, pp. 735–747.
[169] J. Yuan, Closed-loop manipulator control using quaternion feedback, IEEE Journal on Robotics and Automation 4 (1988) 434–440.
[170] R. Yui, R. Hinata, D.N. Nenchev, Upward CoM motion of a humanoid robot resolved in terms of mixed quasi-velocity, Robotic Life Support Laboratory, Tokyo City University, 2017 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00009-2.
[171] X. Yun, N. Sarkar, Unified formulation of robotic systems with holonomic and nonholonomic constraints, IEEE Transactions on Robotics and Automation 14 (1998) 640–650.
[172] J. Zhao, N.I. Badler, Real-Time Inverse Kinematics With Joint Limits and Spatial Constraints, Technical Report MS-CIS-89-09, University of Pennsylvania, 1989.
[173] M. Zucker, S. Joo, M. Grey, C. Rasmussen, E. Huang, M. Stilman, A. Bobick, A general-purpose system for teleoperation of the DRC-HUBO humanoid robot, Journal of Field Robotics 32 (2015) 336–351.