Chapter 2
Redundant Manipulators

2.1 Introduction

In the last decade, redundant manipulators have been the subject of study for many researchers and engineers.[1] While most manipulators have enough DOFs to perform tasks in their end-effector task space, that is, providing any desired pose (position and orientation), their workspace is limited due to mechanical constraints on joints and obstacles that may be present in the work area. Redundant manipulators have extra DOFs compared to the minimum DOFs required for reaching their task space. This allows the redundant manipulators to perform tasks that require high dexterity. They can use the extra DOFs in their benefit to avoid their joint limits and the obstacles in the workspace, while still reaching a desired end-effector pose in the task space.

Avoiding joint limits [70] and obstacles [19, 5] are two of the features of redundant manipulators that have been exploited more often. However, the dexterity of this kind of manipulators can be used to satisfy any desirable kinematic or dynamic characteristic. An example of desired kinematic characteristics is posture control [17], in which the manipulator is programmed to choose a desired set of poses out of all possible poses that can perform a desired task. Examples of desired dynamic characteristic are controlling the contact force of the end-effector [56] or selecting poses that have optimum inertia [71].

The dexterity of redundant manipulators is sometimes comparable to that of a human arm. Redundant manipulators are employed in very important applications where dexterity is required. Perphaps, one of the most famous applications of redundant manipulators is in the International Space Station, where the Special Purpose Dexterous Manipulator (SPDM) (also known as Dexter or Canada Arm 2 in short) is being employed.

Note that the extra DOFs of a redundant manipulator make it kinematically different than a non redundant one. Therefore, the mathematical methods developed

[1] The theories in chapter are strongly based on the second chapter of the book titled: "Control of Redundant Manipulators: Theory and Experiments," by R. V. Patel and F. Shadpey, Springer-Verlag Berlin Heidelberg 2005.

F. Fahimi, *Autonomous Robots*, DOI 10.1007/978-0-387-09538-7_2

for non redundant manipulators are not applicable to a redundant one. Specifically, the "inverse kinematic" problem for a redundant manipulator has multiple solutions in general. Methods that deal with the multiple solutions of the "inverse kinematic" problem for redundant manipulators and can find the best solution that satisfies a desired criteria are known as "redundancy resolution" methods.

In this chapter, first, the kinematics of redundant manipulators is introduced. Then, the most common methods for redundancy resolution are discussed. Finally, the performance of different redundancy resolution methods are studied from two different view points of robustness with respect to algorithmic and kinematic singularity, and flexibility with respect to incorporation of different additional desired tasks, e.g., obstacle avoidance or joint limit avoidance.

2.1.1 Kinematics of Redundant Manipulators

For a manipulator, the task space is the space that defines the pose (position and orientation) of the end-effector. For example, for a manipulator whose end-effector moves in a plane, the end-effector pose can be defined by two position components and one orientation angle. Hence, the task space dimension is three. The joint space for a manipulator is comprised of all the variables that define the configuration of the joints.

For a redundant manipulator, there are more joint variables than there are DOFs for the end-effector. In other words, when the dimension of the task space m for a manipulator is larger than the dimension of the joint space n, the manipulator is said to be redundant.

Normally, the variables that define the pose of the end-effect with respect to a fixed frame of reference are gathered in one single vector as the end-effector pose. Here, this vector is denoted by the $(m \times 1)$ vector $\mathbf{x}$. Also, the variables that define the configuration of the joints are organized in a vector. Here, this vector is named $\mathbf{q}$, (which is $n \times 1$). The difference of the joint space dimension and the task space dimension is called the degree of redundancy, that is, $r = n - m, (r \geq 1)$ is the degree of redundancy.

As one can guess, the pose of the end-effector in space depends on the configuration of the joints. Mathematically, this can be expressed as a functional relation between the end-effector pose vector $\mathbf{x}$ and the joint variables vector $\mathbf{q}$ as

$$\mathbf{x} = \mathbf{f}(\mathbf{q}). \tag{2.1}$$

This relation is known as the forward kinematics relation. Also, the (linear and angular) velocity components for the end-effector can be related to the rate of change of the joint variables. This relation can be expressed in mathematical terms as

$$\dot{\mathbf{x}} = \mathbf{J}_e(\mathbf{q})\dot{\mathbf{q}}, \tag{2.2}$$

where $\dot{\mathbf{x}}$ contains the linear and angular velocity components of the end-effector, and $\mathbf{J}_e$ is the $(m \times n)$ Jacobian of the end-effector. Equation (2.2) is known as the differential kinematics of the manipulator.

Equation (2.2) has an interesting mathematical interpretation. All the possible joint variable velocities $\dot{\mathbf{q}}$ form an $n \times 1$-dimensional mathematical space that is a subset of $\Re^n$. Also, all the possible end-effector velocity vectors $\dot{\mathbf{x}}$ form an $m \times 1$-dimensional mathematical space that is a subset of $\Re^m$. Here, $\Re$ is a set of real numbers. With these definitions, at any fixed $\mathbf{q}$, the Jacobian matrix $\mathbf{J}_e(\mathbf{q})$ can be interpreted as a linear transformation that maps vectors from the space $\Re^n$ into the space $\Re^m$.

Similar to any other linear transformation, the input space $\Re^n$ of the Jacobian matrix has two important associated subspaces. These two subspaces are called the range and the null space (Fig. 2.1). The range of the Jacobian matrix is the subspace of $\Re^n$ that is covered by the transformation. Physically, these are joint velocities that are mechanically possible to be generated by the manipulator's drive mechanism. The range denoted by $\Re(\mathbf{J}_e)$ is mathematically defined by

$$\Re(\mathbf{J}_e) = \{\mathbf{J}_e\dot{\mathbf{q}} | \dot{\mathbf{q}} \in \Re^n\}. \tag{2.3}$$

The null space of the Jacobian matrix is a subset of the input space $\Re^n$ that is mapped to a zero vector in the output space $\Re^m$ by the Jacobian matrix. Physically, these are the achievable joint velocities that do not generate any velocity at the end-effector. The null space of the Jacobian matrix is denoted by $\aleph(\mathbf{J}_e)$ and can be mathematically defined by

$$\aleph(\mathbf{J}_e) = \{\dot{\mathbf{q}} \in \Re^n | \mathbf{J}_e\dot{\mathbf{q}} = \mathbf{0}\}. \tag{2.4}$$

More information about the mathematical definition of the null space can be found in Section A.1.

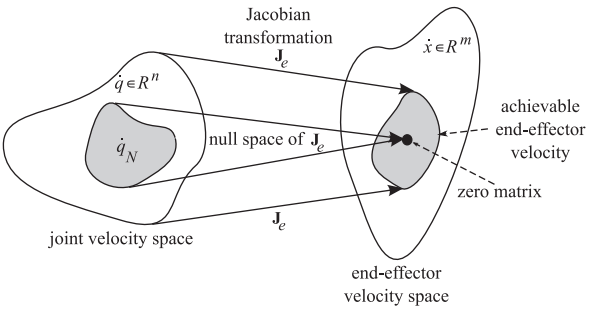

Fig. 2.1 The Jacobian matrix $\mathbf{J}_e$ maps the joint velocity space onto the end-effector velocity space. The null space of the Jacobian matrix $\aleph(\mathbf{J}_e)$ maps a portion of the joint velocity space $\dot{\mathbf{q}}_\aleph$ onto zero end-effector velocity

The existence of the null space, as defined in Eq. (2.4), for the Jacobian matrix is the underlying mathematical basis for redundant manipulators. Physically, Eq. (2.4) implies that the velocities $\dot{\mathbf{q}}_{\aleph}$ picked from the null space $\aleph(\mathbf{J}_e)$ do not generate any velocity $\dot{\mathbf{x}}$ at the end-effector, i.e.,

$$\mathbf{J}_e \dot{\mathbf{q}}_{\aleph} = \mathbf{0}. \tag{2.5}$$

Although the velocities $\dot{\mathbf{q}}_{\aleph}$ do not generate any motion at the end-effector, they generate internal joint motions. Therefore, these velocities can be used to satisfy any requirement that the redundant manipulator must meet, for example, obstacle avoidance for the links, while the end-effector is performing its main task without being disturbed. This can be mathematically described as follows. Consider a desired end-effector velocity $\dot{\mathbf{x}}^d$ that can be generated by applying the joint rates $\dot{\mathbf{q}}^d$. This implies that

$$\dot{\mathbf{x}}^d = \mathbf{J}_e \dot{\mathbf{q}}^d. \tag{2.6}$$

Now, assume that the joint velocities $\dot{\mathbf{q}}_{\aleph}$ are selected from the null space $\aleph(\mathbf{J}_e)$ by an algorithm. The joint velocities $\dot{\mathbf{q}}^d + \alpha \dot{\mathbf{q}}_{\aleph}$, where α is a scalar multiplier, still generate the desired end-effector velocity because

$$\mathbf{J}_e(\dot{\mathbf{q}}^d + \alpha \dot{\mathbf{q}}_{\aleph}) = \mathbf{J}_e \dot{\mathbf{q}}^d + \mathbf{0} = \dot{\mathbf{x}}^d. \tag{2.7}$$

The dimension of the null space from which $\dot{\mathbf{q}}_{\aleph}$'s can be selected depends on the rank of the Jacobian matrix. If the Jacobian matrix $\mathbf{J}_e(\mathbf{q})$ has full column rank (see Section A.2) at a given joint position $\mathbf{q}$, then the dimension of the null space $\aleph(\mathbf{J}_e)$ is equal to the degree of redundancy. If the Jacobian matrix has a rank of $m' < m$, the dimension of $\aleph(\mathbf{J}_e)$ is equal to $(n - m')$.

Since the choice of velocities that belong to the null space is not unique, there are several ways in which the desired main task $\dot{\mathbf{x}}^d$ can be achieved. In other words, there are multiple solutions to the inverse kinematics problem for a redundant manipulator. These multiple solutions can be used wisely to the benefit of the user. To wisely use these multiple solutions, useful additional constraints can be defined. There are two approaches for defining additional constraints: global and local. Global approaches achieve optimal behavior along the whole trajectory which ensures superior performance over local methods [64, 44, 76]. However, their computational burden makes them unsuitable for real-time sensor-based manipulator control applications. For that reason, here, the local approaches, which lead to local optimal behavior, are discussed.

Example 2.1. Consider a planar Prismatic-Revolute-Revolute (PRR) 3-DOF manipulator with joint variables q_1, q_2, and q_3 (Fig. 2.2). The Cartesian coordinates of the end-effector x_1 and x_2 are assumed as the task space with two dimensions. The link lengths for the second and third links are l_2 and l_3, respectively.

(a) Determine the degree of redundancy of this manipulator.
(b) Derive the Jacobian matrix for this manipulator.

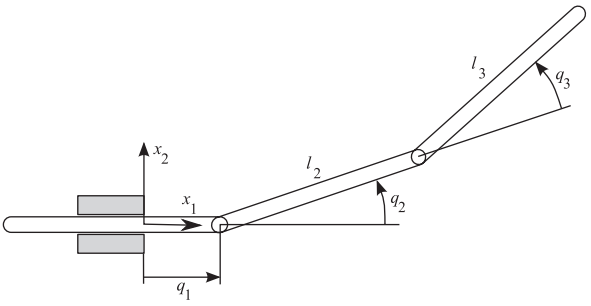

Fig. 2.2 A redundant Prismatic-Revolute-Revolute (PRR) manipulator

Solution. The joint space vector and the task space vector are defined as $\mathbf{q} = [q_1, q_2, q_3]^T$ and $\mathbf{x} = [x_1, x_2]^T$, respectively.

(a) The dimension of the joint space and the task space are $n = 3$ and $m = 2$, respectively. Therefore, the degree of redundancy is $r = n - m = 1$.
(b) To find the Jacobian, first, the position of the end-effector is written as a function of joint parameters. By observing the geometry of the manipulator, one can write

$$x_1 = q_1 + l_2 \cos(q_2) + l_3 \cos(q_2 + q_3), \tag{2.8}$$
$$x_2 = l_2 \sin(q_2) + l_3 \sin(q_2 + q_3), \tag{2.9}$$

which can be written in the matrix form of Eq. (2.1) as

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} q_1 + l_2 \cos(q_2) + l_3 \cos(q_2 + q_3) \\ l_2 \sin(q_2) + l_3 \sin(q_2 + q_3) \end{bmatrix}. \tag{2.10}$$

Differentiating the above equations with respect to time yields

$$\dot{x}_1 = \dot{q}_1 - l_2 \dot{q}_2 \sin(q_2) - l_3(\dot{q}_2 + \dot{q}_3) \sin(q_2 + q_3), \tag{2.11}$$
$$\dot{x}_2 = l_2 \dot{q}_2 \cos(q_2) + l_3(\dot{q}_2 + \dot{q}_3) \cos(q_2 + q_3), \tag{2.12}$$

which can be written in the matrix form of Eq. (2.2) as

$$\dot{\mathbf{x}} = \mathbf{J}_e \dot{\mathbf{q}}, \tag{2.13}$$

where

$$\mathbf{J}_e = \begin{bmatrix} 1 & -l_2 \sin(q_2) - l_3 \sin(q_2 + q_3) & -l_3 \sin(q_2 + q_3) \\ 0 & +l_2 \cos(q_2) + l_3 \cos(q_2 + q_3) & +l_3 \cos(q_2 + q_3) \end{bmatrix}. \tag{2.14}$$

This completes the solution to the example.

2.2 Redundancy Resolution at the Velocity Level

Usually, in the real world applications of manipulators, the desired trajectory (timed position and orientation) of the end-effector is defined as the main task. For the control of the manipulator, however, the trajectory of the joint variables are required. Therefore, the solution to the inverse kinematics problem (more commonly referred to as redundancy resolution for redundant manipulators) is necessary. At the first step of redundancy resolution, the solution is done in the velocity level, that is, the desired joint rates that generate a desired velocity for the end-effector are calculated. This is known as the redundancy resolution at the velocity level.

The reader should be reminded that the redundancy resolution for a redundant manipulator is not trivial. Because of the kinematic redundancy, there are always more unknown joint velocities than there are equations. In this section, the mathematical methods that allow the solution for redundant manipulators at the velocity level are presented and discussed. The mathematical methods can be categorized under two types of exact and approximate solution methods.

2.2.1 Exact Solutions

The pseudo-inverse method and the augmented Jacobian are two exact solution methods that are discussed in this section.

2.2.1.1 Pseudo-Inverse Method

Here, a joint instantaneous velocity $\dot{\mathbf{q}}$ must be found such that it generates a given desired instantaneous velocity $\dot{\mathbf{x}}$ at the end-effector of a redundant manipulator. The instantaneous joint velocities can be found by seeking the exact solution of Eq. (2.2) for $\dot{\mathbf{q}}$ for a given $\dot{\mathbf{x}}$. One of the methods used for obtaining this exact solution is finding the pseudo-inverse of the matrix $\mathbf{J}_e$, denoted by $\mathbf{J}_e^{\dagger}$, and using it as

$$\dot{\mathbf{q}}_p = \mathbf{J}_e^{\dagger}\dot{\mathbf{x}}, \tag{2.15}$$

where the subscript p indicated that this is the primary solution to Eq. (2.2). This solution can be later enhanced by adding solutions $\dot{\mathbf{q}}_{\aleph}$ from the null space of the Jacobian matrix $\mathbf{J}_e$. The pseudo-inverse of $\mathbf{J}_e$ can be written as

$$\mathbf{J}_e^{\dagger} = \mathbf{v}\boldsymbol{\sigma}^{*}\mathbf{u}^{T}, \tag{2.16}$$

where $\boldsymbol{\sigma}$, $\mathbf{v}$, and $\mathbf{u}$ are obtained from the singular-value decomposition (SVD) of $\mathbf{J}_e$ [35], and $\boldsymbol{\sigma}^{*}$ is the transpose of $\boldsymbol{\sigma}$ with all the non-zero values reciprocated (see Section A.3 for more details). Equation (2.16) can also be written in the following summation form

$$\mathbf{J}_e^{\dagger} = \sum_{i=1}^{m'} \frac{1}{\sigma_i} \hat{\mathbf{v}}_i \hat{\mathbf{u}}_i^T, \tag{2.17}$$

where m' is the number of nonzero diagonal components of the matrix $\boldsymbol{\sigma}$, and σ_i is the i-th nonzero diagonal element of the matrix $\boldsymbol{\sigma}$, and $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{u}}_i$ are the i-th column of the matrices $\mathbf{v}$ and $\mathbf{u}$, respectively.

If $\mathbf{J}_e$ has full row rank, then its pseudo-inverse is given by

$$\mathbf{J}_e^{\dagger} = \mathbf{J}_e^T (\mathbf{J}_e \mathbf{J}_e^T)^{-1}. \tag{2.18}$$

The pseudo-inverse method can provide a solution independent of any unbalance in the number of equations and unknowns in Eq. (2.2). In other words, even if the forward kinematic equation is under-specified, square, or over-specified, the pseudo-inverse method can easily specify a solution. However, there are some disadvantages in using this simple method alone.

For example, as mentioned before, Eq. (2.15) only provides the primary solution, which is not in the null space of the Jacobian $\mathbf{J}_e$. This means that the redundancy of the manipulator, which has extra DOFs, cannot be exploited for any useful purpose that could be defined as additional task by a user. This problem can be solved by adding a joint velocity vector $\dot{\mathbf{q}}_{\aleph}$ that belongs to the null space of the Jacobian matrix $\mathbf{J}_e$ to the primary solution as [25]

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_p + \dot{\mathbf{q}}_{\aleph}. \tag{2.19}$$

As shown in Eq. (2.7), the new joint velocity $\dot{\mathbf{q}}$ still satisfies Eq. (2.2). The term $\dot{\mathbf{q}}_{\aleph}$ can be selected as

$$\dot{\mathbf{q}}_{\aleph} = (\mathbf{I} - \mathbf{J}_e^{\dagger} \mathbf{J}_e) \boldsymbol{\nu}, \tag{2.20}$$

where $\boldsymbol{\nu}$ is an arbitrary n-dimensional vector, which will be wisely chosen to satisfy a desired additional task. This desired additional task can be torque and acceleration minimization [70], singularity avoidance [63], or obstacle avoidance [19, 5]. To achieve any of these additional tasks, a cost function can be defined $\Phi(\mathbf{q})$, whose optimum value would ensure the desired additional task. Then, the arbitrary vector $\boldsymbol{\nu}$ can be selected such that the solution given by Eq. (2.19) converges to the optimum value of the cost function. This can be done by choosing the arbitrary vector as

$$\boldsymbol{\nu} = -\nabla \Phi(\mathbf{q}) = -\frac{\partial \Phi}{\partial \mathbf{q}} = -[\frac{\partial \Phi}{\partial q_1} \quad \dots \quad \dots \quad \frac{\partial \Phi}{\partial q_n}]^T. \tag{2.21}$$

Another problem with the solutions provided by Eq. (2.15) is that they may lead to singular configurations for the manipulator, at which the Jacobian matrix $\mathbf{J}_e$ does not have full rank [59]. At those singular configurations, the end-effector of the manipulator cannot generate velocity components in certain directions, which is not

desirable. Close to a singular position, very large joint rates are needed to generate an end-effector velocity in certain directions. Mathematically, for a singular posture, even the largest element of the matrix $\boldsymbol{\sigma}$ is very close to zero. Since the reciprocals of the elements of $\boldsymbol{\sigma}$ appear in the matrix $\boldsymbol{\sigma}^*$ in Eq. (2.16) or (2.17), the joint rates resulting from Eq. (2.15) are very large.

Example 2.2. Consider the PRR redundant manipulator of Example 2.1 (Fig. 2.2). If the link lengths l_2 and l_3 are 0.5 m, find

(a) the joint rates that generate an end-effector velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s, if the arm is at a non singular posture $\mathbf{q} = [0.25\text{ m}, \pi/12\text{ rad}, \pi/3\text{ rad}]^T$;
(b) the joint rates that generate an end-effector velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s, if the arm is at the singular posture $\mathbf{q} = [0.25\text{ m}, \pi/2\text{ rad}, 0\text{ rad}]^T$.

Solution. The Jacobian matrix of Eq. (2.14) derived in Example 2.1 is used in this example.

(a) The Jacobian matrix (2.14) at the non singular posture is

$$\mathbf{J}_e = \begin{bmatrix} 1.0000 & -0.6124 & -0.4830 \\ 0.0000 & 0.6124 & 0.1294. \end{bmatrix} \tag{2.22}$$

As seen from the above matrix, at a non singular posture, the Jacobian matrix has full row rank (the rank of $\mathbf{J}_e = 2$, which is equal to the number of rows of the Jacobian, $n = 2$). In this situation, the expression $\mathbf{J}_e\mathbf{J}_e^T$ is non singular and the pseudo-inverse $\mathbf{J}_e^\dagger$ can be calculated from

$$\mathbf{J}_e^\dagger = \mathbf{J}_e^T(\mathbf{J}_e\mathbf{J}_e^T)^{-1}. \tag{2.23}$$

For $q_1 = 1.25$ m, $q_2 = \pi/12$ rad, and $q_3 = \pi/3$ rad, this yields

$$\mathbf{J}_e^\dagger = \begin{bmatrix} 0.8931 & 0.9974 \\ 0.0634 & 1.6345. \\ -0.3023 & -0.0072 \end{bmatrix} \tag{2.24}$$

Since $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s,

$$\dot{\mathbf{q}}_p = \mathbf{J}_e^\dagger\dot{\mathbf{x}}^d = \begin{bmatrix} 0.4466 \\ 0.0319 \\ -0.1511 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s.} \\ \text{rad/s} \end{matrix} \tag{2.25}$$

(b) At a singular posture, the Jacobian matrix does not have full row rank (rank of $\mathbf{J}_e < n$). Calculating the Jacobian matrix (2.14) confirms that its rank is 1 ($m' = 1$).

$$\mathbf{J}_e = \begin{bmatrix} 1.00 & -1.00 & -0.50 \\ 0.00 & 0.00 & 0.00 \end{bmatrix} \tag{2.26}$$

Since the Jacobian matrix does not have full row rank, the expression $\mathbf{J}_e\mathbf{J}_e^T$ is singular (has no inverse) and the method used in the previous part of this example fails. In such a situation, the pseudo-inverse of the Jacobian matrix must be calculated using the SVD method. In this method, three matrices $\mathbf{u}$, $\boldsymbol{\sigma}$, and $\mathbf{v}$ are found such that $\mathbf{u}\boldsymbol{\sigma}\mathbf{v} = \mathbf{J}_e$.[2]

$$\mathbf{u} = \begin{bmatrix} -1.0000 & 0.0000 \\ 0.0000 & 1.0000 \end{bmatrix} \quad \boldsymbol{\sigma} = \begin{bmatrix} 1.5000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{2.27}$$

$$\mathbf{v} = \begin{bmatrix} -0.6667 & -0.7452 & -0.0000 \\ 0.6667 & -0.5963 & -0.4472 \\ 0.3333 & -0.2981 & 0.8944 \end{bmatrix}. \tag{2.28}$$

The matrix $\boldsymbol{\sigma}^*$ is

$$\boldsymbol{\sigma}^* = \begin{bmatrix} 0.6667 & 0.0000 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}. \tag{2.29}$$

The pseudo-inverse of the Jacobian is calculated as follows

$$\mathbf{J}_e^\dagger = \mathbf{v}\boldsymbol{\sigma}^*\mathbf{u}^T, \tag{2.30}$$

which results in

$$\mathbf{J}_e^\dagger = \begin{bmatrix} 0.4444 & 0.0000 \\ -0.4444 & 0.0000 \\ -0.2222 & 0.0000 \end{bmatrix}. \tag{2.31}$$

Finally the joint rates are

$$\dot{\mathbf{q}}_p = \mathbf{J}_e^\dagger\dot{\mathbf{x}} = \begin{bmatrix} 0.2222 \\ -0.2222 \\ -0.1111 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s.} \\ \text{rad/s} \end{matrix} \tag{2.32}$$

This completes the solution to the example.

[2] The following command in MATLAB calculates the SVD matrices: $[\mathbf{u}, \boldsymbol{\sigma}, \mathbf{v}] = \text{SVD}(\mathbf{J}_e)$.

2.2.1.2 Augmented Jacobian Method

In the augmented Jacobian method [27, 67], for a redundant manipulator with the degree of redundancy of $r = n - m$, r additional tasks are defined. These additional tasks, which are a function of joint variables $\mathbf{q}$, are organized in an $r \times 1$ vector, represented by $\mathbf{z}$. Since the additional task $\mathbf{z}$ is a function of the joint variables vector $\mathbf{q}$, an $r \times n$ Jacobian matrix $\mathbf{J}_c$, known as the Jacobian of the additional task, can be defined that relates their rate of change as

$$\dot{\mathbf{z}} = \mathbf{J}_c \dot{\mathbf{q}}. \tag{2.33}$$

Equation (2.33) adds r equations to the forward kinematics equations (2.2), which brings the total number of equations to n. Since there are n unknown joint rates in $\dot{\mathbf{q}}$, as long as the derivative of the additional task $\dot{\mathbf{z}}$ is defined, the number of equations and unknowns are balanced. This can be mathematically expressed by defining an augmented task vector as

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \tag{2.34}$$

and expressing the augmented task in terms of the joint rate vector as

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{z}} \end{bmatrix} = \mathbf{J}_a \dot{\mathbf{q}}, \tag{2.35}$$

where $\mathbf{J}_a$ is the $(n \times n)$ augmented Jacobian matrix,

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_e \\ \mathbf{J}_c \end{bmatrix}. \tag{2.36}$$

$\mathbf{J}_e$ and $\mathbf{J}_c$ are the $(m \times n)$ and $(r \times n)$ Jacobian matrices of the main and additional tasks, respectively. Once again, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the task vectors of the main, augmented, and additional tasks, respectively.

Since the augmented Jacobian matrix in Eq. (2.35) is square, the solution for the joint rates $\dot{\mathbf{q}}$ can be simply found by using the inverse of $\mathbf{J}_a$. This is a really simple approach, however, there are two major disadvantages associated with this method [71].

Calculation of the inverse of the augmented Jacobian matrix is required in this method. For the inverse of the augmented Jacobian matrix to exist at all times, the additional tasks must be defined at all times. In other words, part-time additional tasks such as obstacle avoidance or joint limit avoidance that are defined based on some conditions that may not exist at all times cannot be used as additional tasks. Hence, this method is not suitable for part-time tasks.

Another disadvantage is that extra singularities can be introduced into the kinematics of the redundant manipulator by defining the additional task. This may be caused by extra singularities that are a consequence of possible rank deficiencies

of the additional task Jacobian $\mathbf{J}_c$ at certain postures. Or this can be caused by an unwanted conflict between the main and the additional task at certain postures, at which the rows of $\mathbf{J}_e$ or $\mathbf{J}_c$ become linearly dependent. This linear dependency, which leads to singularity in the matrix $\mathbf{J}_a$, is task dependent and very hard to predict. In this situation, the solution of Eq. (2.35) based on the inverse of the extended Jacobian $\mathbf{J}_a$ may result in instability near a singular configuration.

Example 2.3. Consider the PRR redundant manipulator of Example 2.1 (Fig. 2.2) at a posture $q_1 = 0.25\,\text{m}$, $q_2 = \pi/12$ rad, and $q_3 = \pi/3$ rad. If the end-effector's angular velocity is defined as an additional task, find the joint rates required to generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s and an angular velocity of $\omega_3^d = \pi/12$ rad/s for the end-effector.

Solution. The additional task is defined based on the desired angular velocity of the end-effector as

$$\dot{z}_1 = \omega_3 = \dot{q}_2 + \dot{q}_3 = \mathbf{J}_c\dot{\mathbf{q}}, \tag{2.37}$$

where the additional task's Jacobian matrix is

$$\mathbf{J}_c = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}. \tag{2.38}$$

The augmented Jacobian for the manipulator becomes

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_e \\ \mathbf{J}_c \end{bmatrix} = \begin{bmatrix} 1 & -l_2\sin(q_2) - l_3\sin(q_2+q_3) & -l_3\sin(q_2+q_3) \\ 0 & +l_2\cos(q_2) + l_3\cos(q_2+q_3) & +l_3\cos(q_2+q_3) \\ 0 & 1 & 1 \end{bmatrix}. \tag{2.39}$$

Substituting the values for q_1, q_2, and q_3 and inverting $\mathbf{J}_a$ yields

$$\mathbf{J}_a^{-1} = \begin{bmatrix} 1 & 0.2979 & 0.4483 \\ 0 & 2.0706 & -0.2679. \\ 0 & -2.0706 & 1.2679 \end{bmatrix} \tag{2.40}$$

The first time derivative of the augmented task $\mathbf{y}$ is

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{z}_1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ \pi/12 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{m/s}\,. \\ \text{rad/s} \end{matrix} \tag{2.41}$$

This results in the following joint rates

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}\dot{\mathbf{y}} = \mathbf{J}_a^{-1}\begin{bmatrix} 0.5 \\ 0.0 \\ \pi/12 \end{bmatrix} = \begin{bmatrix} 0.6174 \\ -0.7010 \\ 0.3319 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s.} \\ \text{rad/s} \end{matrix} \tag{2.42}$$

This completes the solution to the example.

2.2.2 Approximate Solution Methods

2.2.2.1 Singularity Avoidance

Singular postures for a manipulator is not desirable because at a singular posture, the inverse kinematics of a manipulator does not have a meaningful solution. In practice, this means that close to a singular posture, generating a velocity component in certain directions at the end-effector of a manipulator requires very high joint rates, which are not physically possible for the joints to afford. A redundant manipulator can avoid singular postures by exploiting its extra DOFs than that required for a given main task. Here, the use of this feature of redundant manipulators is discussed.

If $\dot{\mathbf{x}}^d$ is the desired main task, the redundancy resolution problem can be formulated as finding the joint rate $\dot{\mathbf{q}}$ that approximately satisfies Eq. (2.2) by minimizing the cost function

$$F = \|\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d\|^2. \tag{2.43}$$

To avoid the singularities, the weighted norm of the joint rates is added to the above cost function. That way, high joint rates are penalized, causing the manipulator not to move close to the singularity posture.

$$F = \|\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d\|^2 + \|\lambda\dot{\mathbf{q}}\|^2 \tag{2.44}$$

This cost function can be expanded in the following form

$$\begin{aligned} F &= (\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d)^T(\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d) + \lambda^2\dot{\mathbf{q}}^T\dot{\mathbf{q}}, \\ &= \dot{\mathbf{q}}^T\mathbf{J}_e^T\mathbf{J}_e\dot{\mathbf{q}} - 2\dot{\mathbf{q}}^T\mathbf{J}_e^T\dot{\mathbf{x}}^d + (\dot{\mathbf{x}}^d)^T\dot{\mathbf{x}}^d + \lambda^2\dot{\mathbf{q}}^T\dot{\mathbf{q}}. \end{aligned} \tag{2.45}$$

The partial derivate of the cost function F with respect to $\dot{\mathbf{q}}^T$ vanishes for $\dot{\mathbf{q}}$ that minimizes F.

$$\frac{\partial F}{\partial \dot{\mathbf{q}}^T} = 2(\mathbf{J}_e^T\mathbf{J}_e\dot{\mathbf{q}} + \lambda^2\dot{\mathbf{q}} - \mathbf{J}_e^T\dot{\mathbf{x}}^d) = 0 \tag{2.46}$$

Solving the partial derivative of the cost function F for the unknown $\dot{\mathbf{q}}$ results in

$$\dot{\mathbf{q}}_\lambda = (\mathbf{J}_e^T\mathbf{J}_e + \lambda^2\mathbf{I})^{-1}\mathbf{J}_e^T\dot{\mathbf{x}}^d. \tag{2.47}$$

The solution to Eq. (2.47), which is unique, closely approximates the exact solution while avoids high joint rates.

One can compare this approximate result with the exact solution by studying the SVD of the exact pseudo-inverse (Eq. 2.17) and that of the coefficient matrix of $\dot{\mathbf{x}}^d$ in Eq. (2.47). The singular values of the exact and the approximate solution are

$$\frac{1}{\sigma_i}, \quad \frac{\sigma_i}{\sigma_i^2 + \lambda^2}, \tag{2.48}$$

respectively. The weight λ is what makes the actual difference between the singular values of the two solutions. Since $\lambda \neq 0$, there are no singularities for the approximate solution. Also, if λ is selected to be small, when the manipulator is far from a singular posture and σ_i's are large, the singular values of the exact and approximate solutions are very close, causing close solutions for $\dot{\mathbf{q}}$. Furthermore, when the manipulator is close to a singular posture, the singular values have the same order as λ. In these cases, the weight λ^2 in the denominator reduces the potentially high norm joint rates.

Example 2.4. Consider the PRR redundant manipulator of Example 2.1. Using the approximate solution method, find the joint rates that generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s for the end-effector when the manipulator is

(a) at a non singular posture of $q_1 = 0.25$ m, $q_2 = \pi/12$ rad, and $q_3 = \pi/3$ rad;
(b) at a singular posture of $q_1 = 0.25$ m, $q_2 = \pi/2$ rad, and $q_3 = 0$ rad.

Solution. Assume $\lambda = 0.1$.

(a) The Jacobian matrix derived in Eq. (2.14) at the non singular posture is

$$\mathbf{J}_e = \begin{bmatrix} 1.0000 & -0.6124 & -0.4830 \\ 0.0000 & 0.6124 & 0.1294 \end{bmatrix} \tag{2.49}$$

The approximate solution is

$$\dot{\mathbf{q}}_p = (\mathbf{J}_e^T \mathbf{J}_e + \lambda^2 \mathbf{I})^{-1} \mathbf{J}_e^T \dot{\mathbf{x}}^d = \begin{bmatrix} 0.4379 \\ 0.0239 \\ -0.1498 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s}, \\ \text{rad/s} \end{matrix} \tag{2.50}$$

which is close to the exact solution obtained in Example 2.1 (Eq. 2.25). The error in the end-effector's velocity introduced due to the approximate solution is

$$\mathbf{e}_{\dot{x}} = \mathbf{J}_e \dot{\mathbf{q}}_p - \dot{\mathbf{x}}^d = \begin{bmatrix} -0.0044 \\ -0.0048 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{m/s}. \end{matrix} \tag{2.51}$$

This error is negligible.

(b) The Jacobian matrix derived in Eq. (2.14) at the singular posture is

$$\mathbf{J}_e = \begin{bmatrix} 1.0000 & -1.0000 & -0.5000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix}. \tag{2.52}$$

The approximate solution is

$$\dot{\mathbf{q}}_p = (\mathbf{J}_e^T \mathbf{J}_e + \lambda^2 \mathbf{I})^{-1} \mathbf{J}_e^T \dot{\mathbf{x}}^d = \begin{bmatrix} 0.2212 \\ -0.2212 \\ -0.1106 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s}, \\ \text{rad/s} \end{matrix} \tag{2.53}$$

which is close to the exact solution obtained in Example 2.1 (Eq. 2.32). The error in the end-effector's velocity introduced due to the approximate solution is

$$\mathbf{e}_{\dot{x}} = \mathbf{J}_e\dot{\mathbf{q}}_p - \dot{\mathbf{x}}^d = \begin{bmatrix} -0.0022 \\ 0.0000 \end{bmatrix} \text{m/s}. \tag{2.54}$$

This error is negligible. Since the posture under consideration is a singular configuration for the PRR manipulator, the method would not result in a solution if λ was assumed to be zero.

2.2.2.2 Configuration Control

The configuration control is another approximate solution method. It is derived by using the same idea of minimization of a cost function as was used for singularity avoidance [69, 70, 68]. In the configuration control method, in addition to the main task $\dot{\mathbf{x}}$, an additional task $\dot{\mathbf{z}}$, and a singularity avoidance task are considered.

$$\dot{\mathbf{x}} = \mathbf{J}_e\dot{\mathbf{q}} \tag{2.55}$$

$$\dot{\mathbf{z}} = \mathbf{J}_c\dot{\mathbf{q}} \tag{2.56}$$

Note that there is no restriction on the dimension of the additional task unlike for the augmented Jacobian method of Section 2.2.1.2.

The desired velocities for the main and additional tasks are defined as $\dot{\mathbf{x}}^d$ and $\dot{\mathbf{z}}^d$, respectively. Then, the joint rates $\dot{\mathbf{q}}$ are found such that the error for the main and the additional tasks are minimized while high joint rates are penalized. To implement this idea, a cost function is defined as follows

$$F = (\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d)^T\mathbf{W}_e(\mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d) + (\mathbf{J}_c\dot{\mathbf{q}} - \dot{\mathbf{z}}^d)^T\mathbf{W}_c(\mathbf{J}_c\dot{\mathbf{q}} - \dot{\mathbf{x}}^d) + \dot{\mathbf{q}}^T\mathbf{W}_v\dot{\mathbf{q}}, \tag{2.57}$$

where $\mathbf{W}_e(m \times m)$, $\mathbf{W}_c(k \times k)$, and $\mathbf{W}_v(n \times n)$ are diagonal positive-definite weighting matrices that assign priority to the main, additional, and singularity avoidance tasks. In Eq. (2.57), the first term penalizes the error in the main task's velocity, the second term penalizes the error in the additional task's velocity, and the third term penalizes high joint rates, hence, causes the manipulator to avoid singularities.

The joint rates that minimize the cost function (2.57) can be found by equating the derivative of F to zero. The derivate of the cost function is

$$\frac{\partial F}{\partial \dot{\mathbf{q}}^T} = 2(\mathbf{J}_e^T\mathbf{W}_e\mathbf{J}_e + \mathbf{J}_c^T\mathbf{W}_c\mathbf{J}_c + \mathbf{W}_v)\dot{\mathbf{q}} - 2(\mathbf{J}_e^T\mathbf{W}_e\dot{\mathbf{x}}^d + \mathbf{J}_c^T\mathbf{W}_c\dot{\mathbf{z}}^d). \tag{2.58}$$

The joint rates are

$$\dot{\mathbf{q}} = (\mathbf{J}_e^T\mathbf{W}_e\mathbf{J}_e + \mathbf{J}_c^T\mathbf{W}_c\mathbf{J}_c + \mathbf{W}_v)^{-1}(\mathbf{J}_e^T\mathbf{W}_e\dot{\mathbf{x}}^d + \mathbf{J}_c^T\mathbf{W}_c\dot{\mathbf{z}}^d). \tag{2.59}$$

Note that there is no restriction on the dimension of the additional task unlike for the augmented Jacobian method of Section 2.2.1.2. Therefore, the disadvantages of the augmented Jacobian method do not exist for the configuration control method. Any part-time additional task, for example, joint limit avoidance or obstacle avoidance, can be defined as the additional task. When the additional task is not active, for example, the joints are not close to their limits, there are not as many active tasks as the degree of redundancy ($k < r$). In these situations, Eq. (2.59) provides a solution similar to that of the singularity avoidance method. When the additional task is active, for example, when some of the joints are close to their limits, the number of active additional tasks can be larger than the degree of redundancy ($k > r$). In those cases, Eq. (2.59) gives the best solution that minimizes the cost function F.

Since there are no hard limitations on the dimension of the additional tasks, the method of configuration control is very powerful and flexible. Any desirable kinematics for the manipulator (such as posture control, joint limit avoidance, and obstacle avoidance [70]), or any dynamic measure of performance that can be formulated as a kinematic function of joint positions or rates (such as contact force, inertia, etc. [71]) can be used as an additional task.

Example 2.5. Consider the PRR redundant manipulator of Example 2.1 and the additional task introduced in Example 2.3. Assume that the main task is three times more important than the additional task and 30 times more important than singularity avoidance. Find the joint rates that generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s and an angular velocity of $\omega_3^d = \pi/12$ rad/s for the end-effector, if the manipulator is at

(a) a non singular posture of $q_1 = 0.25\,\text{m}$, $q_2 = \pi/12$ rad, and $q_3 = \pi/3$ rad;
(b) a singular posture of $q_1 = 0.25\,\text{m}$, $q_2 = \pi/2$ rad, and $q_3 = 0$ rad.

Solution. The dimensions of the main task $\mathbf{x}$, the additional task $\mathbf{z}$, and the joint space are $m = 2$, $k = 1$, and $n = 3$, respectively. The weight matrices are defined as

$$\mathbf{W}_e = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad \mathbf{W}_c = \begin{bmatrix} 1 \end{bmatrix}, \quad \mathbf{W}_v = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \tag{2.60}$$

(b) Using Eq. (2.59) with $\mathbf{J}_e$ (Eq. 2.14) and $\mathbf{J}_c$ (Eq. 2.38) and $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s and $\dot{\mathbf{z}}^d = [\pi/12]$ rad/s results in

$$\dot{\mathbf{q}} = \begin{bmatrix} 0.5764 \\ -0.0283 \\ 0.2338 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s.} \\ \text{rad/s} \end{matrix} \tag{2.61}$$

The errors in the main and augmented tasks are

$$\dot{\mathbf{e}}_e = \mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d = \begin{bmatrix} -0.0192 \\ 0.0129 \end{bmatrix} \text{m/s} \\ \text{m/s'}, \tag{2.62}$$

$$\dot{\mathbf{e}}_c = (\dot{q}_2 + \dot{q}_3) - \dot{z}_1^d = -0.0562 \text{ rad/s}. \tag{2.63}$$

(b) Using Eq. (2.59) with $\mathbf{J}_e$ (Eq. 2.14) and $\mathbf{J}_c$ (Eq. 2.38) and $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s and $\dot{\mathbf{z}}^d = [\pi/12]$ rad/s results in

$$\dot{\mathbf{q}} = \begin{bmatrix} 0.5669 \\ -0.0373 \\ 0.2461 \end{bmatrix} \begin{matrix} \text{m/s} \\ \text{rad/s.} \\ \text{rad/s} \end{matrix} \tag{2.64}$$

The errors in the main and augmented tasks are

$$\dot{\mathbf{e}}_e = \mathbf{J}_e\dot{\mathbf{q}} - \dot{\mathbf{x}}^d = \begin{bmatrix} -0.0189 \\ 0.0000 \end{bmatrix} \text{m/s} \\ \text{m/s'}, \tag{2.65}$$

$$\dot{\mathbf{e}}_c = (\dot{q}_2 + \dot{q}_3) - \dot{z}_1^d = -0.0530 \text{ rad/s}. \tag{2.66}$$

This completes the solution to this example.

2.3 Redundancy Resolution at the Position Level

So far, different methods were studied with which required joint velocities can be determined such that a given velocity can be achieved at the end-effector of a redundant manipulator. These methods by themselves are not able to provide a set of joint positions that result in a desired postion for the end-effector. To find the joint postions required to bring the end-effector to a given position, the joint rates calculated by the redundancy resolution methods at the velocity level must be integrated. In this subsection, the integration procedure is presented.

The mathematical formulation of the redundancy resolution problem in the position level is described as finding $\mathbf{q}$ such that

$$\mathbf{x}^d = \mathbf{f}(\mathbf{q}), \tag{2.67}$$

where $\mathbf{x}^d$ is the desired position of the end-effector. Since this problem is to be solved by integrating the joint velocities, an initial condition is needed. In the physical sense, this initial condition represents the initial posture of the manipulator from which the motion toward the desired position starts. Assume that the initial posture of the manipulator is described by $\mathbf{q}_1$. At this initial posture, the end-effector is located at $\mathbf{x}_1$, where

$$\mathbf{x}_1 = \mathbf{f}(\mathbf{q}_1) \tag{2.68}$$

A path lying in the workspace of the manipulator must be assumed. The path starts from the initial position of the end-effector, $\mathbf{x}_1$, to its desired position, $\mathbf{x}_d$. The simplest assumption for this path is a line segment connecting the two positions in the Euclidean space. This line segment is divided into N smaller line segments for numerical integration. At any integration step, the joint rates required to move the end-effector along this line are calculated. These joint velocities are integrated sequentially until the end-effector reaches the desired point. The integration algorithm follows.

1. Assume an initial posture, $\mathbf{q}_1$, for the manipulator and calculate the initial position of the end-effector, $\mathbf{x}_1$, from Eq. (2.68).
2. Plan a trajectory from $\mathbf{x}_1$ to $\mathbf{x}_{N+1} = \mathbf{x}^d$ with N intervals, and assume the period of the motion, T.
3. Calculate a planned velocity at the interval k that moves the end-effector toward the desired position as

$$\dot{\mathbf{x}}_k = \alpha \frac{\mathbf{x}^d - \mathbf{x}_k}{(N+1-k)\Delta t}, \quad \Delta t = \frac{T}{N}, \tag{2.69}$$

   where α is the deceleration factor greater than 1, which results in faster velocities at the beginning of the motion and slower velocities closer to the desired position.
4. Find the joint rates that generate the planned end-effector velocity at step k. Any of the methods discussed so far can be used for joint rate calculation, for example, the exact redundancy resolution method using a pseudo-inverse Jacobian matrix.

$$\dot{\mathbf{q}}_k = \mathbf{J}_e^{\dagger}(\mathbf{q}_k)\dot{\mathbf{x}}_k \tag{2.70}$$

5. Find $\mathbf{q}_k$ at the next interval by numerically integrating (2.70). Any method of integration can be used, for example the simple method of Euler.

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \dot{\mathbf{q}}_k \Delta t \tag{2.71}$$

6. Find the new end-effector position

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{q}_{k+1}) \tag{2.72}$$

7. Repeat steps 3 to 6 for $k = 1, \ldots, N$.

When the above algorithm runs to the end, when $k = N$, the last step represents $\mathbf{x}_{N+1} = \mathbf{x}^d = \mathbf{f}(\mathbf{q}_{N+1})$. This indicates that the joint posture corresponding to the last step, $\mathbf{q}_{N+1}$, is the solution to the redundancy resolution problem at the position level.

The solution to the redundancy resolution at the position level via integration can be used in two different ways. Note that the above algorithm actually results in the time history of the joint positions and velocities (joint trajectories) that cause the end-effector to move along the planned path from the initial position, $\mathbf{x}_1$, to the desired position, $\mathbf{x}^d$. If the planned path is important to the user, the history of the joint position and velocities derived from this algorithm must be used to control the manipulator on the planned path. If the planned path is not important to the user and only the desired position is important, simpler joint trajectories can be used to move the manipulator joints form the initial posture to the desired posture, $\mathbf{q}_{N+1}$, found at the Nth step of the above algorithm.

Example 2.6. Consider the PRR redundant manipulator of Example 2.1 at an initial posture $\mathbf{q}_1 = [0, 0, 0]^T$. Find the trajectory of the joints required to move the end-effector of the manipulator from its initial position to the desired position $\mathbf{x}^d = [1.25, 0.25]^T$ m. Assume a period of $T = 10$ s for this motion. Use $N = 1,000$ integration intervals and a deceleration factor of $\alpha = 2$.

Solution. The redundancy resolution algorithm presented in this section is used to solve this example. The approximate redundancy resolution method with singularity avoidance (Eq. 2.47 with $\lambda = 0.1$) is used for calculating the joint rates at Step 4 of the algorithm. Application of the algorithm introduced in this section results in the joint trajectories shown in Fig. 2.3. As can be seen in Fig. 2.4, these joint trajectories cause the end-effector of the manipulator to move from the initial position $\mathbf{x}_1 =$

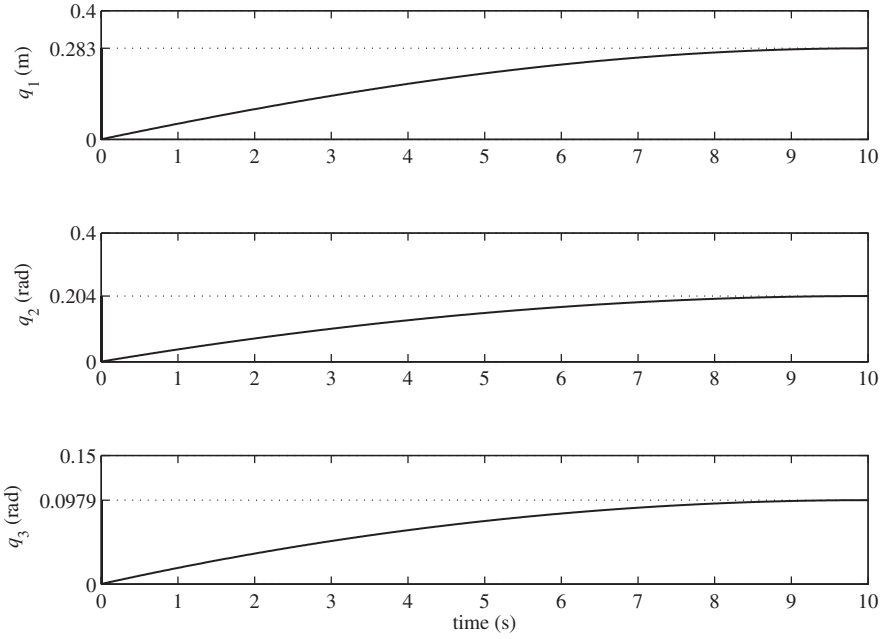

Fig. 2.3 The joint trajectories for the PRR manipulator via the approximate solution method

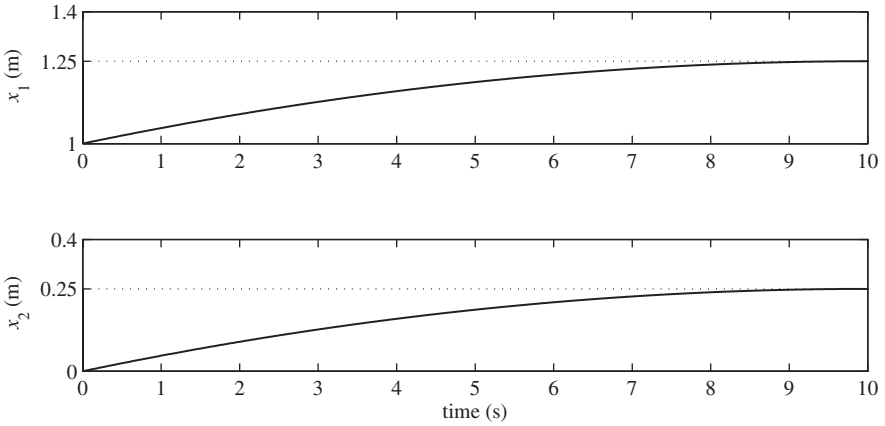

Fig. 2.4 The end-effector trajectory for the PRR manipulator via the approximate solution method

$[1.0, 0.0]^T$ m to the desired position $\mathbf{x}^d = [1.25, 0.25]^T$ m in 10 s. Because of the application of the deceleration factor α, the manipulator slows down when the end-effector gets closer to the desired position. The posture of the manipulator, $\mathbf{q}$, and the position of the end-effector, $\mathbf{x}$, at the end of the motion are

$$\mathbf{q} = \begin{bmatrix} 0.2830 \\ 0.2040 \\ 0.0979 \end{bmatrix} \begin{matrix} \text{m} \\ \text{rad}, \\ \text{rad} \end{matrix} \qquad \mathbf{x} = \begin{bmatrix} 1.2500 \\ 0.2500 \end{bmatrix} \begin{matrix} \text{m} \\ \text{m}. \end{matrix} \tag{2.73}$$

The manipulator at the end of the motion and the path of the end-effector are shown in Fig. 2.5. Since the approximate solution with a small λ (0.1) has been used for redundancy resolution at Step 4 of the algorithm, the path of the end-effector is very close to the planned linear path from the initial to the desired position of the end-effector.

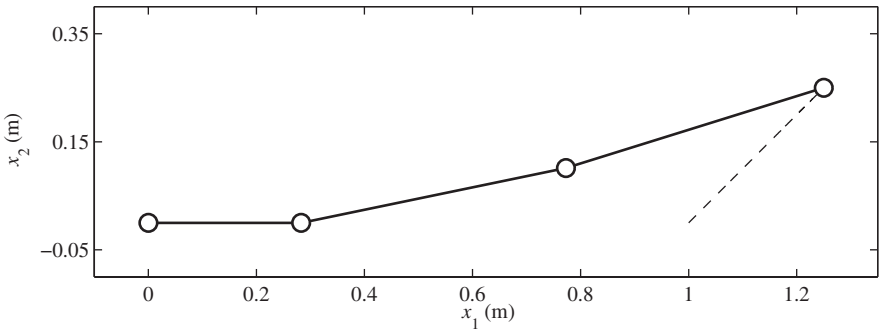

Fig. 2.5 The final posture of the PRR manipulator and the path of the end-effector via the approximate solution method

2.4 Joint Limit Avoidance and Obstacle Avoidance

Joint Limit Avoidance (JLA) and Obstacle Avoidance are two important issues that have to be dealt with when planning the trajectory of the manipulator joints for a certain task at the position level. These two issues can be addressed by using the Configuration Control method, which can serve as a general framework for resolving redundancy. In this section, two general approaches for representing additional tasks are formulated.

2.4.1 Joint Limit Avoidance (JLA)

There are two methods with which the joint limits of a redundant manipulator can be considered when finding joint trajectories that satisfy a given task.

2.4.1.1 Inequality Constraints

In this method, the limits for the joints are defined by part-time constraints as additional tasks. These part-time additional tasks are active for a joint, when the joint position is close to the joint limit. When a joint position is far from the joint limit, the JLA additional task becomes inactive for that joint. In this case, the redundancy can be used for other additional tasks.

A JLA additional task is activated and deactivated by wisely selecting its corresponding weight matrix in the configuration control formulation ($\mathbf{W}_c$ in Eq. (2.59)). It is always a good idea to define a continuous weight for each joint to ensure a smooth joint trajectory. Usually, the weight of the JLA task for a joint is selected to be zero in a region of the joint motion around the center of the joint range. Then, a "buffer" region is assumed with a width τ_i. When the joint position enters this region, the weight of the JLA task is increased from zero to a maximum at the lower limit (q_{imin}) or upper limit ($q_{i\,\max}$). The i-th diagonal entry of the weight matrix $\mathbf{W}_c$ corresponding to joint i is

$$W_{c_{ii}} = \begin{cases} W_0 & \text{if } q_i < q_{i\,\min} \\ \frac{W_0}{2}[1+\cos(\pi(\frac{q_i - q_{i\,\min}}{\tau_i}))] & \text{if } q_{i\,\min} \le q_i \le q_{i\,\min} + \tau_i \\ 0 & \text{if } q_{i\,\min} + \tau_i < q_i < q_{i\,\max} - \tau_i \\ \frac{W_0}{2}[1+\cos(\pi(\frac{q_{i\,\max} - q_i}{\tau_i}))] & \text{if } q_{i\,\max} - \tau_i \le q_i \le q_{i\,\max} \\ W_0 & \text{if } q_i > q_{i\,\max} \end{cases}, \tag{2.74}$$

where W_0 is a user-defined constant representing the coefficient for the weight. This coefficient is normally selected much larger than that of the main task and the singularity avoidance task.

The weight matrix (2.74) is accompanied with an additional task. Since all the joints need to be monitored for the limits, the additional task is defined as a one-to-one function of the joint positions.

$$\mathbf{z} = \mathbf{q} \tag{2.75}$$

With this definition, the corresponding Jacobian for the additional task, $\mathbf{J}_c$, is defined by

$$\mathbf{J}_c = \frac{\partial \mathbf{z}}{\partial \mathbf{q}} = \mathbf{I}. \tag{2.76}$$

Also, since the joint rates must vanish when the joint limits are reached, the desired joint rates when the JLA additional task is active must be selected to be zero.

$$\dot{\mathbf{z}}^d = \mathbf{0} \tag{2.77}$$

The Jacobian for the additional task (2.76) and a weight matrix whose entries are defined by Eq. (2.74) are used with the configuration control method, representred by Eq. (2.59) for joint limit avoidance. An illustrative example follows.

Example 2.7. Consider the redundant PRR manipulator of Example 2.1. Assume a joint limit of $q_{2\max} = 0.1$ rad for the robot's second joint with an activation buffer of $\tau_2 = 0.02$ rad. If the manipulator is at the initial posture of $\mathbf{q}_1 = [0, 0, 0]^T$, find the joint trajectories that cause the end-effector to move to a desired position of $\mathbf{x}^d = [1.25, 0.25]^T$ m.

Solution. Since, in general, it could be assumed that all the joints have a limited range of motion, the additional task, $\mathbf{z}$, contains all the joint variables q_1, q_2, and q_3. Also, the desired additional task rate, $\dot{\mathbf{z}}^d$, for joint limit application is always zero. That is

$$\mathbf{z} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \dot{\mathbf{z}}^d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.78}$$

With the above definition, the Jacobian of the additional task is derived as

$$\mathbf{J}_c = \frac{\partial \mathbf{z}}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.79}$$

The diagonal additional task's weight matrix is chosen such that, in this example, only the second diagonal component is non zero, reflecting the fact that only the second joint is assumed to have limited range of motion.

$$\mathbf{W}_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & W_{c_{22}} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.80}$$

where

$$W_{c_{22}} = \begin{cases} 0 & \text{if } q_2 < 0.08 \\ \frac{50}{2}[1 + \cos(\pi(\frac{0.1 - q_2}{0.02}))] & \text{if } 0.08 \leq q_2 \leq 0.1 \\ 50 & \text{if } q_2 > 0.1 \end{cases}. \tag{2.81}$$

To find the joint trajectories, the algorithm presented in Section 2.3 must be used. However, the above additional task must be incorporated in the algorithm. Furthermore, since the start posture is a singular configuration, a singularity avoidance task must also be incorporated into the algorithm. These are done by replacing the joint rate formula in Step 4 of the algorithm by Eq. (2.59). Since $\dot{\mathbf{z}}^d = [0, 0, 0]^T$, we have

$$\dot{\mathbf{q}}_k = [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \mathbf{J}_c^T \mathbf{W}_c \mathbf{J}_c + \mathbf{W}_v]^{-1} \mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{x}}_k, \tag{2.82}$$

in which the following weight matrices for the main task and the singularity avoidance task are assumed.

$$\mathbf{W}_e = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad \mathbf{W}_v = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \tag{2.83}$$

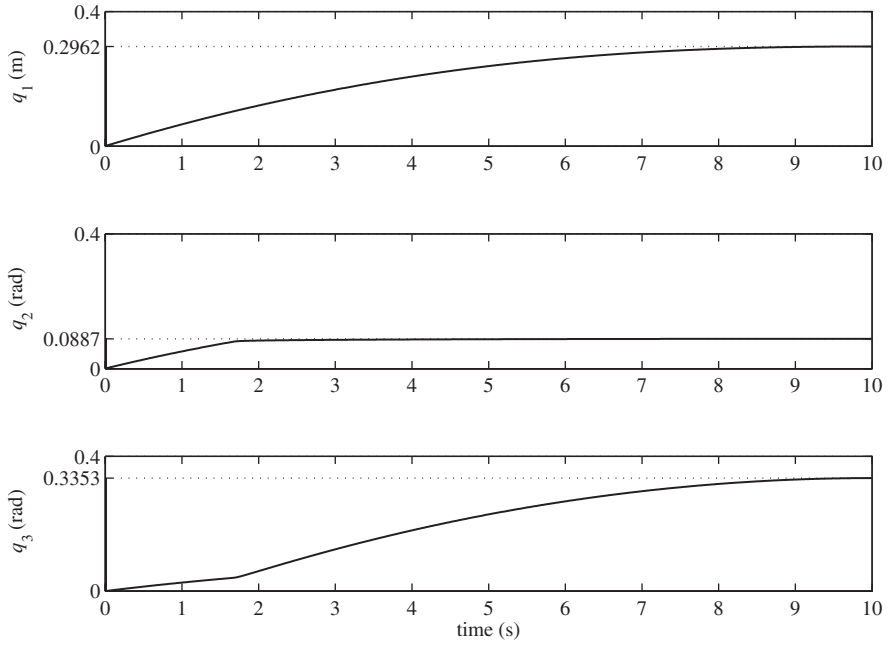

Fig. 2.6 The joint trajectories for the PRR manipulator with JLA via inequality constraints

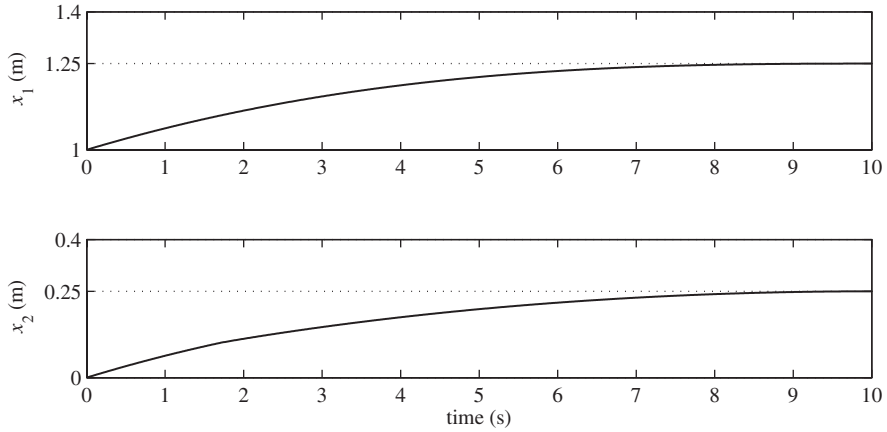

Fig. 2.7 The end-effector trajectory for the PRR manipulator with JLA via inequality constraints

The application of the algorithm in Section 2.3 with the described modifications results in the joint trajectories shown in Fig. 2.6. Because of the application of the joint limit additional task, the second joint stops close to its limit at $q_2 \approx 0.0887$ rad while the other joints continue their motions until the desired end-effector position is achieved. As can be seen in Fig. 2.7, these joint trajectories cause the end-effector of the manipulator to move from the initial position $\mathbf{x}_1 = [1.0, 0.0]^T$ m to the desired position $\mathbf{x}^d = [1.25, 0.25]^T$ m in 10 s. Since a deceleration factor of $\alpha = 2$ has been used, the manipulator slows down when the end-effector gets closer to the desired position. The posture of the manipulator, $\mathbf{q}$, and the position of the end-effector, $\mathbf{x}$, at the end of the motion are

$$\mathbf{q} = \begin{bmatrix} 0.2962 \\ 0.0887 \\ 0.3353 \end{bmatrix} \begin{matrix} \text{m} \\ \text{rad}, \\ \text{rad} \end{matrix} \qquad \mathbf{x} = \begin{bmatrix} 1.2500 \\ 0.2500 \end{bmatrix} \begin{matrix} \text{m} \\ \text{m}. \end{matrix} \tag{2.84}$$

The manipulator at the end of the motion and the path of the end-effector are shown in Fig. 2.8. Since the weighted additional task and singularity avoidance has been introduced at Step 4 of the algorithm for redundancy resolution, the solution for the main task is approximate and the path of the end-effector is not too close to the planned linear path from the initial to the desired position of the end-effector.

2.4.1.2 Kinematic Optimization

In the inequality constraint method, a part-time JLA task was defined. A disadvantage of that method is that the joints move freely until they are close to their limits. There is no sophisticated planning to prevent the joints from reaching their limits. Here, another method for JLA is discussed, in which the JLA task is a full-time task. This method tends to monitor the joints at all times to prevent them from getting close to their limits.

This method takes advantage of the null space of the Jacobian of a redundant manipulator. Normally, an exact solution to the redundancy resolution problem is

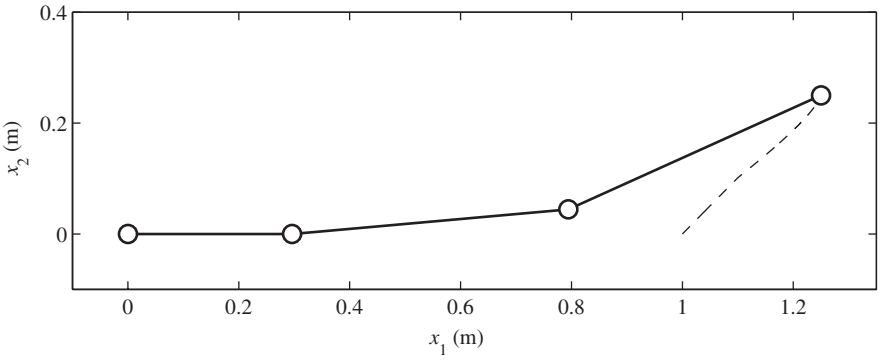

Fig. 2.8 The final posture of the PRR manipulator and the path of the end-effector with JLA via inequality constraints

found using Eq. (2.15). This solution for the joint rates generates the desired motion at the end-effector. Then, the exact joint rate solution is completed by adding a joint rate solution belonging to the null space of the manipulator's Jacobian as shown by Eq. (2.19). The null space solution is found using Eq. (2.20) along with Eq. (2.21). The arbitrary vector $\boldsymbol{\nu}$ in the null space solution comes from optimizing a cost function.

Here, since the goal is to keep the joint far from their limits, a representative of the difference of the position of a joint i to the center q_{c_i} of the joint range Δq_i is defined as a cost function to be minimized. The simplest form of such a function is the quadratic form

$$\Phi(\mathbf{q}) = \sum_{i=1}^{n} \left[\frac{q_i - q_{c_i}}{\Delta q_i} \right]^2 . \tag{2.85}$$

The JLA is now achieved by finding a $\boldsymbol{\nu}$ that optimizes Eq. (2.85), while the end-effector motion is generated by the exact solution. The cost function (2.85) tends to keep the trajectory for the joints around the center of their ranges at all times.

The JLA cost function can be defined with a different view. Similar to the JLA through the inequality constraint method, one may decide to only focus on the joint that is farthest from its center of the range compared to all other joints [48]. This can be translated into the following mathematical relation

$$\Phi(\mathbf{q}) = \max \frac{|q_i - q_{c_i}|}{\Delta q_i} = \| \frac{\mathbf{q} - \mathbf{q}_c}{\Delta \mathbf{q}} \|_\infty, \tag{2.86}$$

which is known as the infinity norm. Although concentrating on the joint closet to its limit using the infinity norm as a cost function is intuitive, it leads to a mathematical complication because the infinity norm is not differentiable. In mathematics, the

p-norm of a vector defined by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{2.87}$$

is an acceptable approximation for the infinity norm. Using the p-norm, a proper cost function for JLA via kinematic optimization can be defined as

$$\Phi(\mathbf{q}) = \| \frac{\mathbf{q} - \mathbf{q}_c}{\Delta \mathbf{q}} \|_p. \tag{2.88}$$

Theoretically, the higher p is, the closer the cost function is to the infinity norm. However, $p = 6$ is sufficient for most practical cases. Also, in some practical cases, avoiding joint limits is more importance for certain joints. In such cases, a weighted version of Eq. (2.88) is used.

$$\Phi(\mathbf{q}) = \|\mathbf{K}(\frac{\mathbf{q} - \mathbf{q}_c}{\Delta \mathbf{q}})\|_p \tag{2.89}$$

where $\mathbf{K}$ is an $n \times n$ diagonal matrix.

Example 2.8. Consider the redundant PRR manipulator of Example 2.1 with a limited joint range for the second joint of $q_2 \in [-0.1, 0.1]$ rad. Use the 2-norm function to incorporate this joint motion limitation in the kinematic redundancy resolution via kinematic optimization. Assume that the manipulator is at the initial posture of $\mathbf{q}_1 = [0, 0, 0]^T$. Find the joint trajectories that cause the end-effector to move to the desired position $\mathbf{x}^d = [1.25, 0.25]^T$ m.

Solution. The 2-norm function for optimization is defined based on Eq. (2.89) as

$$\Phi = \|\mathbf{K}(\frac{\mathbf{q} - \mathbf{q}_c}{\Delta \mathbf{q}})\|_2 = \left(\sum_{i=1}^{3} |K_{ii} \frac{q_i - q_{c_i}}{\Delta q_i}|^2 \right)^{1/2}. \tag{2.90}$$

Since the joint limitation is assumed to exist only for the second joint, K_{11} and K_{33} are selected to be zero. Only K_{22} is assigned a non zero value of 0.01. Furthermore, since the range of the motion of the second joint is $q_2 \in [-0.1, 0.1]$ rad, the center of the range is $q_{c_2} = 0$ and the range is $\Delta q_2 = 0.2$ rad. With these definitions, Eq. (2.90) is reduced to

$$\Phi = |K_{22} \frac{q_2 - q_{c_2}}{\Delta q_2}| = \begin{cases} K_{22} \frac{q_2 - q_{c_2}}{\Delta q_2} & \text{if } q_2 \geq q_{c_2} \\ K_{22} \frac{q_{c_2} - q_2}{\Delta q_2} & \text{if } q_2 < q_{c_2} \end{cases}. \tag{2.91}$$

The gradient of this cost function is needed for taking advantage of the null space of the manipulator's Jacobian to incorporate the joint limit via kinematic optimization. It is calculated as

$$\boldsymbol{v} = -\nabla\Phi = -[\frac{\partial\Phi}{\partial q_1}, \frac{\partial\Phi}{\partial q_2}, \frac{\partial\Phi}{\partial q_3}]^T = [0, -\frac{\partial\Phi}{\partial q_2}, 0]^T, \tag{2.92}$$

where

$$\frac{\partial\Phi}{\partial q_2} = \begin{cases} \frac{K_{22}}{\Delta q_2} & \text{if } q_2 \geq q_{c_2} \\ -\frac{K_{22}}{\Delta q_2} & \text{if } q_2 < q_{c_2} \end{cases}. \tag{2.93}$$

The algorithm presented in Section 2.3 must be used for redundancy resolution in the position level and finding the required joint trajectories. Note that the Step 4 of the presented algorithm, at which the joint rates are calculated, must be modified to address the joint limit avoidance via kinematic optimization. The joint rates required to perform the main task, $\dot{\mathbf{q}}_p$, are calculated via the approximate solution (Eq. 2.47)

$$\dot{\mathbf{q}}_p = (\mathbf{J}_e^T\mathbf{J}_e + \lambda^2\mathbf{I})^{-1}\mathbf{J}_e^T\dot{\mathbf{x}}^d, \tag{2.94}$$

which also includes singularity avoidance. A null space solution, $\dot{\mathbf{q}}_{\aleph}$ from Eq. (2.20), is added to the main task solution. This null space solution does not affect the end-effector's motion, however, it implements the joint limit requirement.

$$\dot{\mathbf{q}}_{\aleph} = (\mathbf{I} - \mathbf{J}_e^{\dagger}\mathbf{J}_e)\boldsymbol{v}, \tag{2.95}$$

where the pseudo-inverse of the Jacobian in the above equation is calculated via approximate solution for singularity avoidance, that is,

$$\mathbf{J}_e^{\dagger} = (\mathbf{J}_e^T\mathbf{J}_e + \lambda^2\mathbf{I})^{-1}\mathbf{J}_e^T. \tag{2.96}$$

Finally, the joint rates that replace Step 4 of the position level redundancy resolution algorithm are

$$\dot{\mathbf{q}}_k = (\mathbf{J}_e^T\mathbf{J}_e + \lambda^2\mathbf{I})^{-1}\mathbf{J}_e^T\dot{\mathbf{x}}_k + (\mathbf{I} - \mathbf{J}_e^{\dagger}\mathbf{J}_e)\boldsymbol{v}. \tag{2.97}$$

The application of the algorithm in Section 2.3 with the described modifications results in the joint trajectories shown in Fig. 2.9. Because of the application of the joint limit via kinematic optimization, the second joint operates well within its limits $q_2 \in [-0.1, 0.1]$ rad. Also, since the optimization criterion is active during the whole motion, the trajectory of the joints are smoother compared to the previous example, in which the joint limit criterion is only active in the "buffer" zone. As can be seen in Fig. 2.10, these joint trajectories cause the end-effector of the manipulator to move from the initial position $\mathbf{x}_1 = [1.0, 0.0]^T$ m to the desired position $\mathbf{x}^d = [1.25, 0.25]^T$ m in 10 s. Since a deceleration factor of $\alpha = 3$ has been used, the manipulator slows down when the end-effector gets closer to the desired position. The posture of the manipulator, $\mathbf{q}$, and the position of the end-effector, $\mathbf{x}$, at the end of the motion are

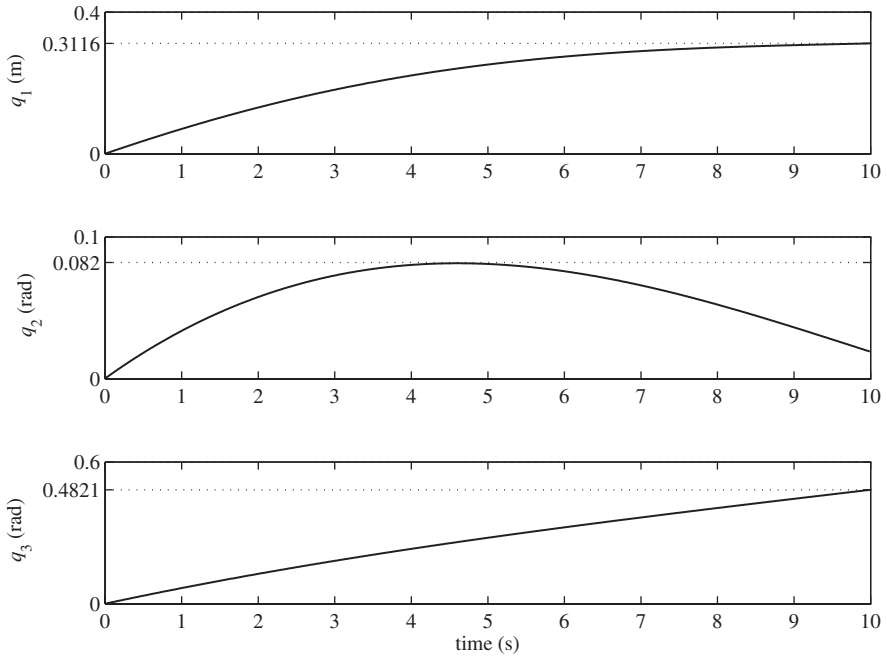

Fig. 2.9 The joint trajectories for the PRR manipulator with JLA via kinematic optimization

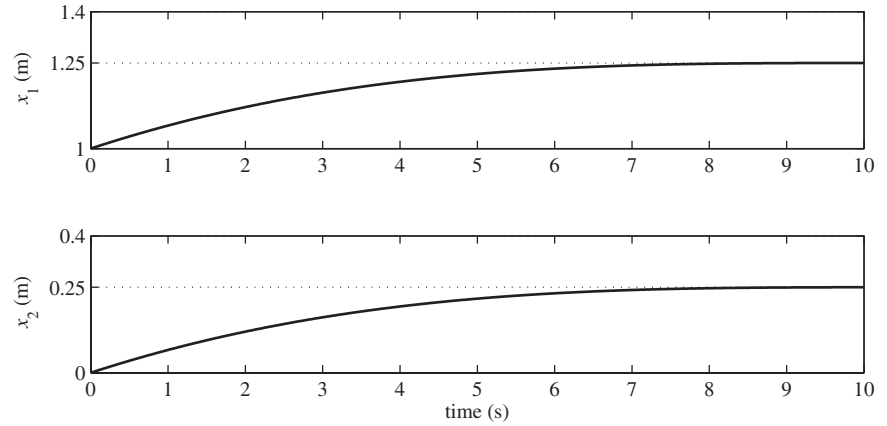

Fig. 2.10 The end-effector trajectory for the PRR manipulator with JLA via kinematic optimization

$$\mathbf{q} = \begin{bmatrix} 0.3116 \\ 0.0193 \\ 0.4821 \end{bmatrix} \begin{matrix} \text{m} \\ \text{rad}, \\ \text{rad} \end{matrix} \qquad \mathbf{x} = \begin{bmatrix} 1.2500 \\ 0.2500 \end{bmatrix} \begin{matrix} \text{m} \\ \text{m}. \end{matrix} \tag{2.98}$$

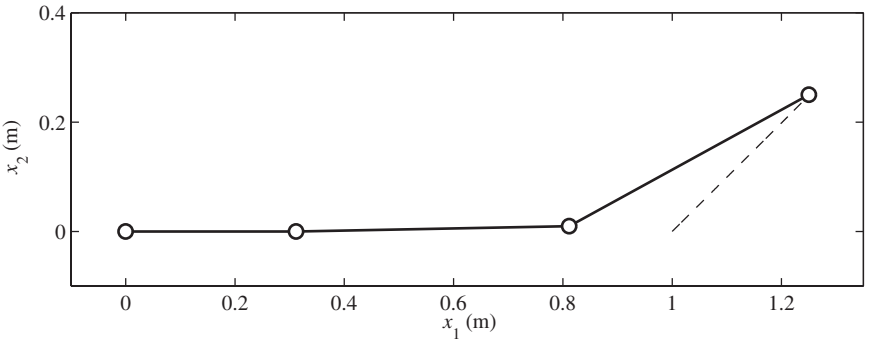

Fig. 2.11 The final posture of the PRR manipulator and the path of the end-effector with JLA via kinematic optimization

The manipulator at the end of the motion and the path of the end-effector are shown in Fig. 2.11. Since the singularity avoidance factor λ has been selected very small (0.1) and the joint limit avoidance has been incorporated as a null space solution, the joints rates calculated at Step 4 of the algorithm for redundancy resolution are very close to the exact solution. As a consequence, the path of the end-effector is very close to the linear path planned from the initial to the desired position of the end-effector.

2.4.2 Obstacle Avoidance

In this section, an outline of an obstacle avoidance algorithm for the 2D workspace of a planar redundant manipulator, applicable to both static and moving obstacles, is given.

2.4.2.1 Algorithm Description

Obstacle avoidance, similar to JLA, is a part-time task, which is only activated when a possibility of collision is detected. The configuration control method, which is useful when part-time additional tasks are involved, is used here for redundancy resolution.

The obstacle avoidance algorithm has three layers [19]. First, the distance of all the manipulators' links to the obstacles must be calculated at any given time. Second, for each link, a decision must be made based on the distance of the link to the obstacle to determine if the obstacle avoidance additional task must be activated for the link. And third, the configuration control method must be used for redundancy resolution to find the joint rates that avoid collision of the links with obstacles.

The distance of a link to an obstacle is calculated through some simplifying assumptions (Fig. 2.12). The links of the planar manipulator are considered as straight

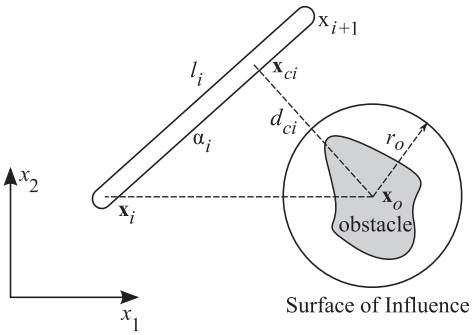

Fig. 2.12 Determination of the critical point and the distance between a link and a Surface of Influence

lines connecting the center of joint coordinate frames. Obstacles are enclosed in circles with diameters larger than the largest obstacle dimension. The thickness of the manipulator links should also be added to the radii of these circles. These circles are called the Surface of Influence (SOI).

With these assumptions, the calculation of the location of the potential point of collision, also known as the critical point, becomes rather simple. The calculation procedure for finding the location of the critical point follows.

If link i is modeled by a line from joint i at the Cartesian coordinates $\mathbf{x}_i$ to joint $i+1$ at the Cartesian coordinates $\mathbf{x}_{i+1}$, the unit vector representing the direction of the link with length l_i is

$$\hat{\mathbf{e}}_i = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{l_i}. \tag{2.99}$$

Assume that the center of the SOI is at the Cartesian coordinates $\mathbf{x}_o$. The projection of a line from joint i to the center of the SOI on the link i is

$$\alpha_i = \hat{\mathbf{e}}_i^T(\mathbf{x}_o - \mathbf{x}_i), \tag{2.100}$$

which represents the dot product of the unit vector $\hat{\mathbf{e}}_i$ and the vector $\mathbf{x}_o - \mathbf{x}_i$ in the matrix notation. Now, the Cartesian coordinates of the critical point can be calculated as

$$\mathbf{x}_{c_i} = \mathbf{x}_i + \alpha_i \hat{\mathbf{e}}_i. \tag{2.101}$$

The distance of the critical point with the center of the SOI is

$$d_{c_i} = \|\mathbf{x}_{c_i} - \mathbf{x}_o\|. \tag{2.102}$$

Based on this distance, the unit vector pointing from the critical point to the center of the obstacle is determined.

$$\hat{\mathbf{u}}_i = \frac{\mathbf{x}_{c_i} - \mathbf{x}_o}{d_{c_i}}, \tag{2.103}$$

or

$$d_{c_i} = \hat{\mathbf{u}}_i^T (\mathbf{x}_{c_i} - \mathbf{x}_o). \tag{2.104}$$

All the links and SOIs are considered, and a critical distance d_{ci} is calculated for each. If, for a link i, this critical distance is smaller than the radius of the SOI (or $r_o - d_{ci} > 0$), the obstacle avoidance additional task for that link is activated. For each link i, the obstacle avoidance additional task can be defined as the normal distance of the link to the SOI. That is

$$z_i = f_i(\mathbf{q}, t) = r_o - d_{c_i}. \tag{2.105}$$

Using Eq. (2.104), one can write the derivative of the additional task as

$$\dot{z}_i = -\frac{d}{dt}(d_{c_i}) = -\hat{\mathbf{u}}_i^T \left(\frac{\partial \mathbf{x}_{c_i}}{\partial \mathbf{q}} \dot{\mathbf{q}} - \dot{\mathbf{x}}_o \right), \tag{2.106}$$

where $\dot{\mathbf{x}}_o$ is the velocity of the center of the SOI, or, in other words, the obstacle [2].

The obstacle avoidance additional task must be defined such that a link i does not enter the SOI of its corresponding obstacle. Therefore, when the additional task becomes active, the desired value for the additional task is

$$z_i^d = 0, \tag{2.107}$$

which also implies that

$$\dot{z}_i^d = \ddot{z}_i^d = 0. \tag{2.108}$$

The Jacobian of the obstacle avoidance additional task must be calculated. Since each link i has its own unique condition regarding the obstacles, each row i of the Jacobian matrix for the additional task is calculated separately based on the position of the critical point on link i. The i-th row of the Jacobian is derived by observing Eq. (2.106).

$$\mathbf{J}_{c_i} = -\hat{\mathbf{u}}_i^T \mathbf{J}_{\mathbf{x}_{c_i}}, \tag{2.109}$$

where

$$\mathbf{J}_{\mathbf{x}_{c_i}} = \frac{\partial \mathbf{x}_{c_i}}{\partial \mathbf{q}}. \tag{2.110}$$

The Jacobian of the additional task is assembled from $\mathbf{J}_{c_i}$ of each link that is facing a SOI.

Example 2.9. Assume an obstacle at the Cartesian coordinate $(0.6, 0.0)$ m in the workspace of the redundant PRR manipulator of Example 2.1. Assume a Surface of Influence with radius $r_0 = 0.15$ m for the obstacle. The manipulator is at an initial posture of $\mathbf{q}_1 = [0, 0.5, 0.5]^T$. The end-effector must reach the desired position of $\mathbf{x}^d = [1.15, 0.15]^T$ m in 10 s. Determine the joint trajectories for the redundant manipulator.

Solution. To simplify the solution to this example, it is assumed that only link 2 may collide with the obstacle. First, the distance of the critical point on link 2 with the center of the SOI is calculated and an additional task is defined based on this distance. Then, the Jacobian of the additional task is determined. Finally, this augmented task is incorporated into the redundancy resolution via the configuration control method. The details of the solution procedure follow.

The Cartesian coordinates of joints 2 and 3 are written as

$$\mathbf{x}_2 = \begin{bmatrix} q_1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} q_1 + l_2 \cos q_2 \\ l_2 \sin q_2 \end{bmatrix}. \tag{2.111}$$

The unit vector representing the direction of link 2, $\hat{\mathbf{e}}_2$, and the distance of the critical point with joint 2, α_2, can be derived as

$$\hat{\mathbf{e}}_2 = \frac{\mathbf{x}_3 - \mathbf{x}_2}{l_2} = \begin{bmatrix} \cos q_2 \\ \sin q_2 \end{bmatrix}, \tag{2.112}$$

$$\alpha_2 = \hat{\mathbf{e}}_2^T (\mathbf{x}_o - \mathbf{x}_2). \tag{2.113}$$

The Cartesian coordinates of the critical point, $\mathbf{x}_{c_2}$, and the distance of the critical point to the center of the SOI, d_{c_i}, are

$$\mathbf{x}_{c_2} = \mathbf{x}_2 + \alpha_2 \hat{\mathbf{e}}_2 = \begin{bmatrix} q_1 + \alpha_2 \cos q_2 \\ \alpha_2 \sin q_2 \end{bmatrix}, \tag{2.114}$$

$$d_{c_i} = \|\mathbf{x}_{c_2} - \mathbf{x}_o\| = \sqrt{(x_{c_2,1} - x_{o,1})^2 + (x_{c_2,2} - x_{o,2})^2}. \tag{2.115}$$

The unit vector pointing from the critical point to the obstacle center is

$$\hat{\mathbf{u}}_2 = \frac{\mathbf{x}_{c_2} - \mathbf{x}_o}{d_{c_2}}. \tag{2.116}$$

The Jacobian of the critical point is calculated as

$$\mathbf{J}_{\mathbf{x}_{c_2}} = \frac{\partial \mathbf{x}_{c_2}}{\partial \mathbf{q}} = \begin{bmatrix} 1 & -\alpha_2 \sin q_2 & 0 \\ 0 & \alpha_2 \cos q_2 & 0 \end{bmatrix}. \tag{2.117}$$

The second column of the Jacobian of the additional task, which corresponds to obstacle avoidance for link 2, is

$$\mathbf{J}_{c_2} = \begin{cases} [0, 0, 0] & \text{if } d_{c_2} > r_o \\ -\hat{\mathbf{u}}_2^T \mathbf{J}_{\mathbf{x}_{c_2}} & \text{if } d_{c_2} \leq r_o \end{cases}. \tag{2.118}$$

This means that, the additional task corresponding to link 2 is only active when the distance of the critical point to the center of the SOI, d_{c_2}, is smaller than the radius of the SOI for the obstacle, r_o.

Now, the Jacobian for the additional task for the whole manipulator, $\mathbf{J}_c$, must be assembled using the augmented tasks corresponding to each link. Note that the first and the third row of the Jacobian of the additional task, $\mathbf{J}_c$, have all zero components, because no obstacle avoidance is being considered for links 1 and 3 in this example for simplicity. The above note dictates that the Jacobian of the additional task, $\mathbf{J}_c$, becomes

$$\mathbf{J}_c = \begin{bmatrix} 0 & 0 & 0 \\ J_{c_2,1} & J_{c_2,2} & J_{c_2,3} \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.119}$$

The Step 4 of redundancy resolution algorithm the position level presented in section 2.3 must be modified such that the incorporation of the additional task is possible. This is done by using Eq. (2.59). Since the desired rate for the additional task, $\dot{\mathbf{z}}^d$, is zero, Eq. (2.59) is reduced to

$$\dot{\mathbf{q}}_k = [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \mathbf{J}_c^T \mathbf{W}_c \mathbf{J}_c + \mathbf{W}_v]^{-1} \mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{x}}_k. \tag{2.120}$$

The following weight matrices are assumed

$$\mathbf{W}_e = \mathbf{I}, \quad \mathbf{W}_c = (1000)\mathbf{I}, \quad \mathbf{W}_v = (0.1)\mathbf{I}. \tag{2.121}$$

where $\mathbf{I}$ is a 3×3 identity matrix. The application of the algorithm in Section 2.3 with the described modifications results in the joint trajectories shown in Fig. 2.13. Because of the implementation of the obstacle avoidance for link 2 via configuration control method, the second link stops far from the obstacle. Also, in this example, the deceleration factor α was chosen equal to 3.0, which makes the joint speeds much slower toward the end of the motion compared to the joint speeds at the beginning of the motion. As can be seen in Fig. 2.14, these joint trajectories cause the end-effector of the manipulator to move from its initial position to the desired position $\mathbf{x}^d = [1.15, 0.15]^T$ m in 10 s. The posture of the manipulator, $\mathbf{q}$, and the position of the end-effector, $\mathbf{x}$, at the end of the motion are

$$\mathbf{q} = \begin{bmatrix} 0.1841 \\ 0.3668 \\ -0.4254 \end{bmatrix} \begin{matrix} \text{m} \\ \text{rad}, \\ \text{rad} \end{matrix} \qquad \mathbf{x} = \begin{bmatrix} 1.1500 \\ 0.1500 \end{bmatrix} \begin{matrix} \text{m} \\ \text{m}. \end{matrix} \tag{2.122}$$

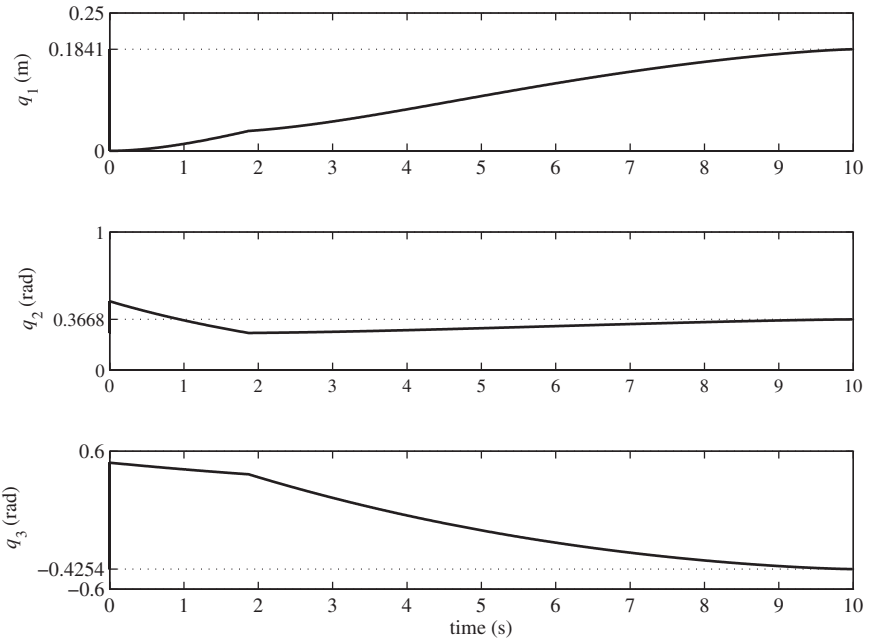

Fig. 2.13 The joint trajectories for the PRR manipulator with obstacle avoidance via configuration control

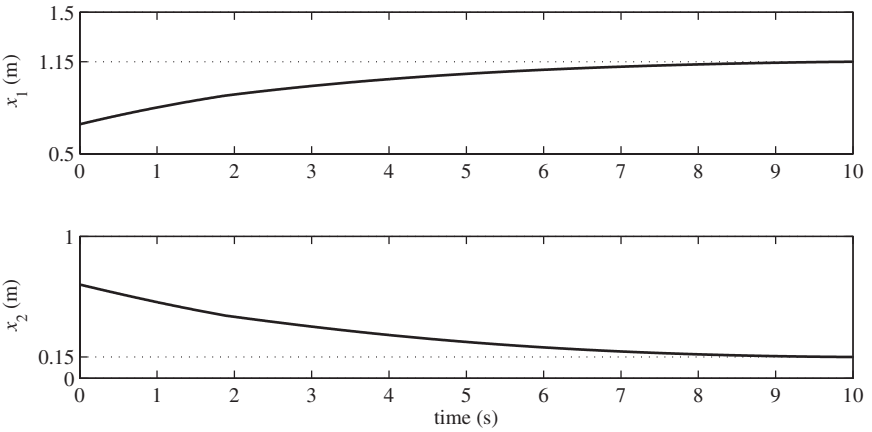

Fig. 2.14 The end-effector trajectory for the PRR manipulator with obstacle avoidance via configuration control

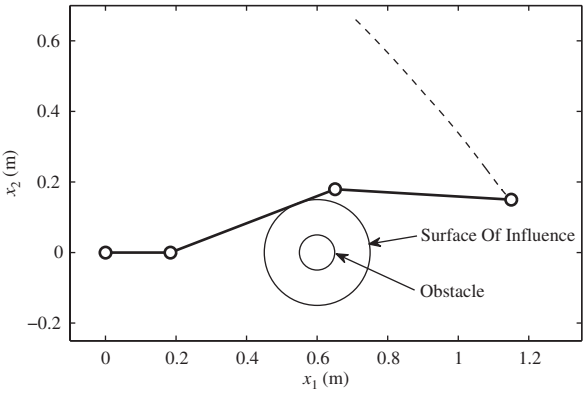

Fig. 2.15 The final posture of the PRR manipulator and the end-effector path with obstacle avoidance via configuration control

The manipulator at the end of the motion and the path of the end-effector are shown in Fig. 2.15. Since the obstacle avoidance has been implemented via the configuration control method, the solution to joint rates used in Step 4 of the algorithm (based on Eq. 2.59) is a trade-off between the main and the additional tasks, with a higher weight for the additional task. As a consequence, the path of the end-effector is far from the linear path planned from the initial to the desired position of the end-effector.

2.5 Summary

In this chapter, the basic issues needed for the analysis of kinematically redundant manipulators were presented. Different redundancy resolution schemes were reviewed. The formulation of the additional tasks to be used by the redundancy resolution module were presented in this chapter. Joint limit avoidance, which is one of the most useful additional tasks, was studied in detail. The basic formulation of static and moving obstacle collision avoidance task in 2D workspace was presented.

Problems

Problem 2.1. Consider the 3DOF planar Revolute-Revolute-Prismatic (RRP) manipulator shown in Fig. 2.16 with joint variables q_1, q_2, and q_3. The Cartesian coordinates of the end-effector x_1 and x_2 are assumed as the task space with two dimensions. The link lengths for the first, second, and third links are l_1, l_2, and l_3, respectively.

(a) Determine the degree of redundancy of this manipulator.
(b) Derive the Jacobian matrix for this manipulator.

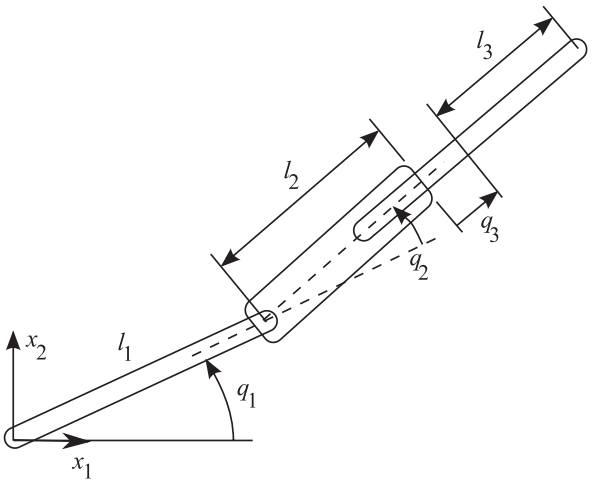

Fig. 2.16 A RRP manipulator

Problem 2.2. Consider the RRP redundant manipulator of Problem 2.1. If the link length l_1 is 1.0 m and the lengths l_2 and l_3 are 0.5 m, find

(a) the joint rates that generate an end-effector velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s, if the arm is at a posture $\mathbf{q} = [\pi/12 \text{ rad}, \pi/3 \text{ rad}, 0.25 \text{ m}]^T$;
(b) the joint rates that generate an end-effector velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s, if the arm is at a posture $\mathbf{q} = [\pi/2 \text{ rad}, 0 \text{ rad}, 0.25 \text{ m}]^T$.

Problem 2.3. Consider the RRP redundant manipulator of Problem 2.1 at a posture $q_1 = \pi/6$ rad, $q_2 = \pi/12$ rad, and $q_3 = 0.25$ m. If the end-effector's angular velocity is defined as an additional task, find the joint rates required to generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s and an angular velocity of $\omega_3^d = \pi/12$ rad/s for the end-effector.

Problem 2.4. Consider the RRP redundant manipulator of Problem 2.1. Using the approximate solution method, find the joint rates that generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s for the end-effector when the manipulator is

(a) at a posture of $q_1 = \pi/12$ rad, $q_2 = \pi/3$ rad, and $q_3 = 0.25$ m.
(b) at a posture of $q_1 = \pi/2$ rad, $q_2 = 0$ rad, and $q_3 = 0.25$ m.

Problem 2.5. Consider the RRP redundant manipulator of Problem 2.1 and the additional task introduced in Problem 2.3. Assume that the main task is three times more important than the additional task and 30 times more important than singularity avoidance. Find the joint rates that generate a velocity of $\dot{\mathbf{x}}^d = [0.5, 0.0]^T$ m/s

and an angular velocity of $\omega_3^d = \pi/12$ rad/s for the end-effector, if the manipulator is at

(a) a posture of $q_1 = \pi/3$ rad, $q_2 = \pi/12$ rad, and $q_3 = 0.25$ m;
(b) a posture of $q_1 = \pi/2$ rad, $q_2 = 0$ rad, and $q_3 = 0$ m.

Problem 2.6. Consider the 3DOF planar RRP manipulator of Problem 2.1. The joint variables are q_1, q_2, and q_3. The Cartesian coordinates of the end-effector x_1 and x_2 are assumed as the main task space with two dimensions. The link lengths are $l_1 = 1$ m and $l_2 = l_3 = 0.5$ m.

Assume a joint limit of $q_{3\,\min} = -0.3$ m and $q_{3\,\max} = 0.3$ m for the third joint with an activation buffer of $\tau_3 = -0.02$ m. If the manipulator is at the initial posture of $\mathbf{q}_1 = [-0.86, 0.15, 0.00]^T$, find the joint trajectories that cause the end-effector to move to a desired position of $\mathbf{x}^d = [1.78, 1.42]^T$ m. Plot the components of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\mathbf{x}$, $\dot{\mathbf{x}}$, and the path of the end-effector.

(a) Use the Inequality Constraint method.
(b) Use the Kinematic Optimization method.
(c) Discuss the difference in the obtained plots in part (a) and (b).

Problem 2.7. Assume an obstacle at the Cartesian coordinate $(1.25, 0.0)$ m in the workspace of the redundant RRP manipulator of Problem 2.1. Assume a SOI with radius $r_0 = 0.15$ m for the obstacle. The manipulator is at an initial posture of $\mathbf{q}_1 = [\pi/2, 0, 0]^T$. The end-effector must reach the desired position of $\mathbf{x}^d = [2, 0]^T$ m in 15 s. Determine the joint trajectories for the redundant manipulator.