3 Cartesian Impedance Control: The Rigid Body Case

In this chapter the classical theory of impedance control for rigid body robots is described. The presentation in Section 3.1 is based on the seminal work of Hogan about the concept of impedance control [Hog85a, Hog85b, Hog85c] and on the *Operational Space Formulation* by Khatib [Kha87]. In Section 3.2 the case of a desired impedance is analyzed, in which the inertial behavior must not be shaped explicitly. This brings about the problem of designing the damping matrix in an appropriate way. Furthermore, some additional aspects concerning singularity avoidance, the choice of coordinates, and the stiffness design are discussed.

This chapter refers to the rigid body part of the robot model without considering joint elasticities. The presented topics serve as a prerequisite for the design of Cartesian impedance controllers for the flexible joint robot model from Section 2.2.3 and can readily be combined with the controllers from Chapter 5, 6, and 7.

3.1 Complete Decoupling

Throughout this chapter the flexibility of the joints is neglected. While in the previous chapter the joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ were related to the motor side positions $\boldsymbol{\theta}$ and the link angles $\boldsymbol{q}$ via $\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q})$, they are now considered as the control inputs. The joint angles are summarized in the vector $\boldsymbol{q} \in \mathcal{Q}^p$. By using the notation of Chapter 2 the considered dynamical model of the robot is given by[1]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \ , \tag{3.1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Coriolis/centrifugal matrix, $\boldsymbol{g}(\boldsymbol{q})$ is the vector of gravity torques, and $\boldsymbol{\tau}_{ext}$ is the vector of external torques.

[1] Notice that the joint torques $\boldsymbol{\tau}$, not the motor torques $\boldsymbol{\tau}_m$, are considered here as the control inputs for the rigid body model. Therefore, the respective inertia matrix is $\boldsymbol{M}(\boldsymbol{q})$ instead of $\boldsymbol{M}(\boldsymbol{q}) + \boldsymbol{B}$.

C. Ott: Cartesian Imped. Cntrl. of Redund. & Flexible-Joint Robots, STAR 49, pp. 29–44, 2008.
springerlink.com

3.1.1 Task Formulation

The goal of impedance control is to realize a particular desired dynamical relationship between the robot motion and the external torques. In case of the Cartesian impedance control problem, this relationship is specified in terms of coordinates which describe the motion of the end-effector. In general, the actual configuration of the end-effector can be represented in special coordinates by a homogeneous matrix (see Section 2.1), i.e. as an element of $SE(3)$, which can be computed based on the product of exponentials formula (2.3). For the purpose of controller design, instead, a minimal representation in terms of m end-effector coordinates $\boldsymbol{x} \in \mathbb{R}^m$ is often preferred. In case that all degrees-of-freedom of the end-effector motion are considered in the task, one has $m = 6$. In the following it is assumed that the relationship between these Cartesian coordinates $\boldsymbol{x}$ and the configuration coordinates $\boldsymbol{q} \in \mathcal{Q}$ is given by a known function $\boldsymbol{f} : \mathcal{Q} \to \mathbb{R}^m$, i.e. $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$. In Section 3.5 some more details about possible choices for the Cartesian coordinates $\boldsymbol{x}$ are given.

Throughout this chapter only the non-redundant case is considered, for which the number of joint angles n and the number of Cartesian coordinates m are the same $m = n$. Possible extensions to the redundant case will be treated in the next chapter.

For the formulation of the desired dynamic behavior in terms of the Cartesian coordinates $\boldsymbol{x}$, also the first and the second time derivatives, $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$, are considered. With the Jacobian[2] $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ these derivatives can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \tag{3.2}$$

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ . \tag{3.3}$$

Considering (2.3), one can see that the Cartesian coordinates $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$ depend on the joint variables q_i by trigonometric (for revolute joints) and affine (for prismatic joints) terms. Therefore, one can conclude that the maximum possible singular value $\sigma_{M,J}$ of the Jacobian keeps bounded[3] within $\mathcal{Q}^p$, i.e.

$$\sigma_{M,J} := \sup_{\boldsymbol{q} \in \mathcal{Q}^p} \sigma_{max}(\boldsymbol{J}(\boldsymbol{q})) < \infty \ . \tag{3.4}$$

Besides the restriction to the non-redundant case, which was already mentioned before, it will, furthermore, be assumed that the Jacobian $\boldsymbol{J}(\boldsymbol{q})$ is non-singular[4]. In general $\boldsymbol{J}(\boldsymbol{q})$ will, of course, not be non-singular in the whole configuration space $\mathcal{Q}^p$. Then the analysis of this chapter is restricted to an area $\bar{\mathcal{Q}}^p$ in which the invertibility is ensured. This is established herein by requiring that the minimum possible singular value of the Jacobian

$$\sigma_{m,J} := \inf_{\boldsymbol{q} \in \bar{\mathcal{Q}}^p} \sigma_{min}(\boldsymbol{J}(\boldsymbol{q})) \tag{3.5}$$

[2] In the robotics literature this matrix is sometimes called *analytical Jacobian*, in contrast to the geometrical Jacobian, or the body Jacobian, respectively.

[3] Note that $\mathcal{Q}^p$, as defined in (2.36), describes a subset of the configuration space $\mathcal{Q}$ in which the prismatic joint variables keep bounded.

[4] An appropriate singularity treatment will be briefly described in Section 3.4.

must be bigger than some value $\sigma_0 > 0$. Moreover, it is assumed that the mapping $\boldsymbol{f}(\boldsymbol{q})$ is one-to-one in $\bar{\mathcal{Q}}^p$. The corresponding area in Cartesian coordinates, i.e. the image of $\bar{\mathcal{Q}}^p$ through $\boldsymbol{f}$, is denoted by $\bar{\mathcal{Q}}^p_c$

$$\bar{\mathcal{Q}}^p := \{\boldsymbol{q} \in \mathcal{Q}^p \mid \sigma_{min}(\boldsymbol{J}(\boldsymbol{q})) > \sigma_0 \text{ and } \boldsymbol{f}(\boldsymbol{q}) \text{ is one-to-one}\} \ , \tag{3.6}$$

$$\bar{\mathcal{Q}}^p_c := \boldsymbol{f}(\bar{\mathcal{Q}}^p) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \exists \boldsymbol{q} \in \bar{\mathcal{Q}}^p \ , \ \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{x}\} \ . \tag{3.7}$$

In the set $\bar{\mathcal{Q}}^p_c$ the Cartesian coordinates $\boldsymbol{x}$ can be used as generalized coordinates of the system (3.1). In general it is, of course, not possible to find Cartesian coordinates, for which $\bar{\mathcal{Q}}^p_c$ corresponds to the complete state space $\mathbb{R}^n$. Still, for the analysis of a Cartesian controller it is often interesting to investigate if a (local) stability statement holds also globally under the assumption $\bar{\mathcal{Q}}^p_c = \mathbb{R}^n$. If this is true, the stability region is only restricted by the particular choice of coordinates.

The external torques $\boldsymbol{\tau}_{ext}$ shall be related to the vector of generalized external forces $\boldsymbol{F}_{ext}$ via $\boldsymbol{\tau}_{ext} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_{ext}$. Therefore, $\boldsymbol{F}_{ext}$ is the dual variable to $\dot{\boldsymbol{x}}$ and hence the power which is exchanged between the robot and its environment is given by $\boldsymbol{\tau}_{ext}^T \dot{\boldsymbol{q}} = \boldsymbol{F}_{ext}^T \dot{\boldsymbol{x}}$.

In order to specify the desired impedance behavior the position error $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d$ between $\boldsymbol{x}$ and a (possibly time-varying) *virtual equilibrium position*[5] $\boldsymbol{x}_d$ is introduced. Then the control objective is to achieve a dynamical relationship of the form

$$\boldsymbol{\Lambda}_d \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d \tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext} \tag{3.8}$$

between $\tilde{\boldsymbol{x}}$ and $\boldsymbol{F}_{ext}$, where $\boldsymbol{K}_d$, $\boldsymbol{D}_d$, and $\boldsymbol{\Lambda}_d$ are the symmetric and positive definite matrices of the desired stiffness, damping, and inertia, respectively. In principle one could of course also consider a more general impedance behavior. But in most robotic applications the restriction of the impedance controller to a desired behavior in form of such a *mass-spring-damper-system* is sufficient.

3.1.2 Robot Model in Task Coordinates

The robot model (3.1) is written in joint coordinates $\boldsymbol{q}$, while the desired behavior (3.8) is defined in task coordinates $\boldsymbol{x}$. For the controller design it is easier to rewrite the model (3.1) also in task coordinates as it is done in the operational space formulation [Kha87]. Substituting $\ddot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}})$ from (3.3) and $\boldsymbol{\tau}_{ext} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_{ext}$ into equation (3.1) leads to

$$\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1} \left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \right) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_{ext} \ .$$

With $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{x}}$ from (3.2) substituted in the second and the third term and by pre-multiplying the resulting equation by $\boldsymbol{J}(\boldsymbol{q})^{-T}$, one gets

$$\begin{aligned} &\boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1}\ddot{\boldsymbol{x}} + \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{x}} - \\ &\quad \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{J}}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{x}} + \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{\tau} + \boldsymbol{F}_{ext} \ . \end{aligned}$$

[5] As already mentioned in the introduction the set-point in impedance control usually is called a *virtual* equilibrium point, because it actually should only be reached in case that no external forces act on the robot.

This equation can now be written in the form

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{\tau} + \boldsymbol{F}_{ext} \ , \tag{3.9}$$

where the matrices $\boldsymbol{\Lambda}(\boldsymbol{x})$ and $\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})$ are given by

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1} \ , \tag{3.10}$$

$$\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{J}}(\boldsymbol{q})\right)\boldsymbol{J}(\boldsymbol{q})^{-1} \ , \tag{3.11}$$

with $\boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x})$ and $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{f}^{-1}(\boldsymbol{x}))\dot{\boldsymbol{x}}$. From a control point of view the use of both the joint variables $\boldsymbol{q}$ and the Cartesian coordinates $\boldsymbol{x}$ simultaneously in (3.10) and (3.11) is a little bit misleading, because the considered state variables are only $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ throughout this chapter. A *direct* representation of some components in terms of $\boldsymbol{q}$ on the other hand is sometimes much clearer and simpler[6]. Whenever the two representations are mixed, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ can be thought of being replaced by $\boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x})$ and $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{f}^{-1}(\boldsymbol{x}))\dot{\boldsymbol{x}}$, respectively.

In analogy to the external torques also the gravity torques $\boldsymbol{g}(\boldsymbol{q})$ and the joint torques $\boldsymbol{\tau}$ can be rewritten in form of the equivalent task space gravity forces $\boldsymbol{F}_g(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{g}(\boldsymbol{q})$ and the new input vector $\boldsymbol{F}_\tau$, which is related to $\boldsymbol{\tau}$ via $\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{F}_\tau$. Therefore, the system equations finally have the form

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{x}) = \boldsymbol{F}_\tau + \boldsymbol{F}_{ext} \ . \tag{3.12}$$

The matrices $\boldsymbol{\Lambda}(\boldsymbol{x})$ and $\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})$ are the inertia matrix and the *Coriolis/centrifugal matrix* with respect to the coordinates $\boldsymbol{x}$.

Before the classical impedance control law for the model (3.12) is formulated, two important lemmata of the model (3.12) are presented, which follow directly from Property 2.5 and Property 2.6 (see [dWSB96]).

Lemma 3.1. *The matrix $\boldsymbol{\Lambda}(\boldsymbol{x})$, as defined in (3.10), is symmetric and positive definite for all positions $\boldsymbol{x} \in \bar{\mathcal{Q}}_c^p$.*

Proof. Due to the restriction of $\boldsymbol{x}$ to $\bar{\mathcal{Q}}_c^p$, the matrix $\boldsymbol{\Lambda}(\boldsymbol{x})$ is well defined. The symmetry and the positive definiteness of $\boldsymbol{\Lambda}(\boldsymbol{x})$ follow directly from its definition in (3.10) together with Property 2.5.

Lemma 3.2. *The matrix $\dot{\boldsymbol{\Lambda}}(\boldsymbol{x}) - 2\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})$, with $\boldsymbol{\Lambda}(\boldsymbol{x})$ and $\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})$ as defined in (3.10) and (3.11), is skew symmetric for all $\boldsymbol{x} \in \bar{\mathcal{Q}}_c^p$ and all $\dot{\boldsymbol{x}} \in \mathbb{R}^m$.*

Proof. Due to the restriction of $\boldsymbol{x}$ to $\bar{\mathcal{Q}}_c^p$ both matrices $\boldsymbol{\Lambda}(\boldsymbol{x})$ and $\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})$ are well defined. By Lemma A.22, one has to show that the equality $\dot{\boldsymbol{\Lambda}}(\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})^T$ holds. In the remaining part of the proof the arguments of the matrices will be dropped in order to simplify the notation. From (3.10) and (3.11) and

[6] Since the joint angles $\boldsymbol{q}$ and not the Cartesian coordinates $\boldsymbol{x}$ are the measured quantities a representation in terms of $\boldsymbol{q}$ is also required for the actual implementation.

the symmetry of $\boldsymbol{M}$ (Property 2.5) it follows that the matrices $\dot{\boldsymbol{\Lambda}}$ and $\boldsymbol{\mu}+\boldsymbol{\mu}^T$ are given by

$$\begin{aligned}\dot{\boldsymbol{\Lambda}} &= \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}) \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-T}\dot{\boldsymbol{M}}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-T}\boldsymbol{M}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-1} \ , \\ \boldsymbol{\mu}+\boldsymbol{\mu}^T &= \boldsymbol{J}^{-T}(\boldsymbol{C}+\boldsymbol{C}^T)\boldsymbol{J}^{-1} - \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}\dot{\boldsymbol{J}}\boldsymbol{J}^{-1} \\ &\quad -\boldsymbol{J}^{-T}\dot{\boldsymbol{J}}^T\boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1} \ .\end{aligned}$$

Considering the equality $\dot{\boldsymbol{M}} = \boldsymbol{C}+\boldsymbol{C}^T$, which follows from Property 2.6 together with Lemma A.22, the term $\dot{\boldsymbol{\Lambda}} - \boldsymbol{\mu} - \boldsymbol{\mu}^T$ results in

$$\begin{aligned}\dot{\boldsymbol{\Lambda}} - \boldsymbol{\mu} - \boldsymbol{\mu}^T &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-T}\boldsymbol{M}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-1} \\ &\quad + \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}\dot{\boldsymbol{J}}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-T}\dot{\boldsymbol{J}}^T\boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1} \\ &= (\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-T} + \boldsymbol{J}^{-T}\dot{\boldsymbol{J}}^T\boldsymbol{J}^{-T})\boldsymbol{M}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-T}\boldsymbol{M}(\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-1}\dot{\boldsymbol{J}}\boldsymbol{J}^{-1}) \ .\end{aligned}$$

From the equality

$$\begin{aligned}\boldsymbol{0} &= \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{I})\boldsymbol{J}^{-1} = \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{J}^{-1}\boldsymbol{J})\boldsymbol{J}^{-1} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{J}^{-1} + \boldsymbol{J}^{-1}\dot{\boldsymbol{J}}\boldsymbol{J}^{-1}\end{aligned}$$

one can conclude $\dot{\boldsymbol{\Lambda}} - \boldsymbol{\mu} - \boldsymbol{\mu}^T = \boldsymbol{0}$ which completes the proof.

Notice that these properties are not surprising at all, since (3.12) is nothing else than the model (3.1) written in another set of coordinates. The matrix $\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ in (3.12), however, could also have been chosen differently from (3.11). The special form in (3.11) has the advantage that it ensures the validity of Lemma 3.2. Equation (3.11) corresponds to the transformation of the Christoffel symbols [Fra97, Boo03] from (2.30) into Cartesian coordinates, which means that the same form of $\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ as in (3.11) is obtained, if it is computed via the Christoffel symbols of $\boldsymbol{\Lambda}(\boldsymbol{x})$. Equation (3.10) clearly is the classical coordinate transformation for a covariant tensor of rank 2 [Fra97, Boo03].

The restriction to $\bar{\mathcal{Q}}_c^p$ ensures also that, similar to Property 2.7, the eigenvalues of the Cartesian inertia matrix keep bounded from above and below by some non-zero bounds.

Property 3.3. *Within $\bar{\mathcal{Q}}_c^p$ the eigenvalues $\lambda_i(\boldsymbol{\Lambda}(\boldsymbol{x}))$ of the Cartesian inertia matrix keep bounded, i.e.*

$$0 < \frac{\lambda_{m,M}}{\sigma_{M,J}^2} \leq \lambda_i(\boldsymbol{\Lambda}(\boldsymbol{x})) \leq \frac{\lambda_{M,M}}{\sigma_{m,J}^2} < \infty \quad i = 1, \cdots, n, \ \forall \boldsymbol{x} \in \bar{\mathcal{Q}}_c^p \ ,$$

with $\lambda_{m,M}$ and $\lambda_{M,M}$ as the minimum and maximum possible eigenvalues of the inertia matrix as defined in (2.37) and (2.38), and $\sigma_{m,J}$ and $\sigma_{M,J}$ as the *minimum and maximum possible singular value of the Jacobian as defined in (3.4) and (3.5).*

Proof. This property follows directly from the definition of $\boldsymbol{\Lambda}(\boldsymbol{x})$ in (3.10) together with Property 2.7 and the definition of $\bar{\mathcal{Q}}_c^p$.

This property will be of interest for the analysis of the desired impedance behavior in Section 3.2 as well as for the controller design in Chapter 6.

3.1.3 Classical Impedance Controller

The classical impedance control law [Hog85a, Hog85b] can be directly computed from equation (3.12) [Kha87]. The control input $\boldsymbol{F}_\tau$ which leads to the desired closed loop system (3.8) is given by

$$\begin{aligned}\boldsymbol{F}_\tau = &\boldsymbol{F}_g(\boldsymbol{x}) + \boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}} - \boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1}\left(\boldsymbol{K}_d\tilde{\boldsymbol{x}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}}\right) + \\ &\left(\boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1} - \boldsymbol{I}\right)\boldsymbol{F}_{ext} \ .\end{aligned}$$

This Cartesian impedance controller is then actually implemented via the joint torques $\boldsymbol{\tau}$ as follows

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{F}_\tau && (3.13)\\ &= \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T\left(\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}}\right) - \\ &\quad \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1}\left(\boldsymbol{K}_d\tilde{\boldsymbol{x}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}}\right) + \\ &\quad \boldsymbol{J}(\boldsymbol{q})^T\left(\boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1} - \boldsymbol{I}\right)\boldsymbol{F}_{ext} \ . && (3.14)\end{aligned}$$

One can see that the shaping of the desired impedance in this case also contains a feedback of the external forces $\boldsymbol{F}_{ext}$. These forces are usually measured by means of a force-torque-sensor mounted at the end-effector. But in general there might also be external forces which do not act on the tool but directly on the robot structure and thus cannot be measured. Since these forces are not included in the measurement of $\boldsymbol{F}_{ext}$, it is clear that the closed loop impedance behavior with respect to these forces will be quite different. Notice that, from a practical point of view, this is typically much more relevant in service robotics than in industrial applications. Clearly, when the robot is interacting with humans this interaction is not necessarily restricted to the tool of the robot.

The need for feedback of the external forces follows from the requirement that not only the stiffness and damping behavior of the robot should be shaped but also the inertial behavior. However, in many applications this is not necessary. Therefore, a simplified control objective is considered in the next section, in which an explicit shaping of the apparent inertia is not considered and the emphasis is laid on the shaping of the stiffness and the damping.

3.2 Avoidance of Inertia Shaping

The feedback of external forces $\boldsymbol{F}_{ext}$ can be avoided when the desired inertia $\boldsymbol{\Lambda}_d$ is identical to the robot inertia $\boldsymbol{\Lambda}(\boldsymbol{x})$

$$\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}(\boldsymbol{x}) \ . \qquad (3.15)$$

Since the desired inertia depends on the position $\boldsymbol{x}$, also the relevant centrifugal and Coriolis terms should be considered in the specification of the desired closed loop behavior. This is necessary in order to fulfill Lemma 3.2 and thus to ensure the passivity of the system for the regulation case (see also Proposition 3.5 below). The desired dynamic relationship between $\tilde{\boldsymbol{x}}$ and $\boldsymbol{F}_{ext}$ is given as

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\tilde{\boldsymbol{x}}} + (\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{D}_d)\,\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext}\ , \tag{3.16}$$

where $\boldsymbol{K}_d$ and $\boldsymbol{D}_d$ are again the symmetric and positive definite matrices of desired stiffness and desired damping, respectively. In the regulation case (i.e. for $\dot{\boldsymbol{x}}_d = \boldsymbol{0}$) this control objective is often called *compliance control* problem.

Before presenting the controller which leads to the closed loop system (3.16), a short comparison to the original desired dynamics (3.8) shall be given in order to justify the choice of (3.16). Consider first the case of free motion, i.e. $\boldsymbol{F}_{ext} = \boldsymbol{0}$. The asymptotic stability of the original desired dynamics (3.8) is ensured for this case simply by the fact that the desired stiffness, damping, and inertia matrices are positive definite. The original desired dynamics is even linear and time-invariant. The stability properties of the new dynamics (3.16) are not that obvious. Clearly, the system (3.16) is nonlinear due to the use of a position dependent inertia matrix. Furthermore, it is time-varying[7], since the virtual equilibrium position $\boldsymbol{x}_d(t)$ may be a two-times continuously differentiable function of the time t. Some relevant definitions and lemmata regarding the stability analysis of time-varying systems can be found in Appendix A.1.

Without going into the details, it should be mentioned that the stability of (3.16) for the case of free motion can be shown by considering the (time-varying) Lyapunov function

$$V(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}}, t) = \frac{1}{2}\dot{\tilde{\boldsymbol{x}}}^T\boldsymbol{\Lambda}(\boldsymbol{x})\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{K}_d\tilde{\boldsymbol{x}}\ . \tag{3.17}$$

The change of this Lyapunov function along the solutions of (3.16) is negative semi-definite, which implies stability but not asymptotic stability. However, for free motion, the dynamics in (3.16) is equivalent to the closed loop system of the well known *PD+ controller* [PP88] written in task coordinates. Asymptotic stability of the PD+ controller in configuration space was shown in [PP88]. Furthermore, a strict Lyapunov function for this system was presented in [SK97b]. In both works the boundedness of the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$, as formulated in Property 2.7, was required. The equivalent statement for the Cartesian inertia matrix $\boldsymbol{\Lambda}(\boldsymbol{x})$ is ensured in Property 3.3 due to the restriction to $\bar{\mathcal{Q}}_c^p$. According to the proof in [SK97b] (for the same controller in configuration space) the stability statement formulated in the following proposition holds even globally, if $\bar{\mathcal{Q}}_c^p$ corresponds to the complete state space $\mathbb{R}^n$.

Proposition 3.4. *Let the desired trajectory $\boldsymbol{x}_d(t)$ be continuously differentiable twice. Assume further that the Cartesian coordinates are valid globally, i.e. $\bar{\mathcal{Q}}_c^p = \mathbb{R}^n$. Then for $\boldsymbol{F}_{ext} = \boldsymbol{0}$ the system (3.16) with symmetric and positive definite matrices $\boldsymbol{K}_d$ and $\boldsymbol{D}_d$ is uniformly globally asymptotically stable.*

[7] Notice the occurrence of both $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d(t)$ in (3.16).

Another important feature of the original desired dynamics (3.8) is that in the regulation case (i.e. $\dot{\boldsymbol{x}}_d = \mathbf{0}$) it represents a passive mapping from external generalized forces $\boldsymbol{F}_{ext}$ to the velocity $\dot{\boldsymbol{x}}$. This property is especially important if the interaction of the robot with passive[8] environments is considered. By considering (3.17) as a storage function it can be shown for the regulation case that the system (3.16) also represents a passive mapping from the external force $\boldsymbol{F}_{ext}$ to the velocity $\dot{\boldsymbol{x}}$.

Proposition 3.5. *For $\dot{\boldsymbol{x}}_d(t) = \mathbf{0}$, the system (3.16) with symmetric and positive definite matrices $\boldsymbol{K}_d$ and $\boldsymbol{D}_d$ is time-invariant and represents a passive mapping from the external force $\boldsymbol{F}_{ext}$ to the velocity $\dot{\boldsymbol{x}}$.*

By following the same argumentation as for the classical impedance controller from the last section, one gets the control law

$$\boldsymbol{F}_\tau = \boldsymbol{F}_g(\boldsymbol{x}) + \boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}}_d - \boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} \ ,$$

to achieve the dynamics (3.16). For the implementation of this controller a formulation at the joint level

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_\tau \\ &= \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T \left(\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}}_d - \boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} \right) \end{aligned} \quad (3.18)$$

is more convenient.

The desired stiffness usually is given by the application. The new desired impedance behavior (3.16) then brings up the problem of how to choose the damping matrix. This problem is treated in the next section.

3.3 Design of the Damping Matrix

In case of the original desired dynamics (3.8) the desired inertia matrix $\boldsymbol{\Lambda}_d$ is constant and can be designed in principle such that its eigenvectors coincide with the eigenvectors of $\boldsymbol{K}_d$. The design of the damping matrix can then be reduced to the design of the damping coefficients for n decoupled linear and time-invariant second order systems, each describing the dynamics of the system along one of the eigenvectors.

For the new desired dynamics (3.16) the situation is different. The desired stiffness matrix must be constant and is usually defined by the application being considered. The design of the desired damping matrix, instead, is not so clear. Notice that a constant and diagonal matrix $\boldsymbol{D}_d$, in general, is not a good choice, since the inertia matrix is non-diagonal and time-varying. Instead, one should also take into account the particular structure and the change of $\boldsymbol{\Lambda}(\boldsymbol{x})$ during movement.

For the design of the damping matrix it should be mentioned that all the stability and passivity properties from above will also hold, when the constant

[8] Here the velocity $\dot{\boldsymbol{x}}$ is considered as the input and the negative external force $-\boldsymbol{F}_{ext}$ as the output of the environment.

matrix $\boldsymbol{D}_d$ in (3.16) is replaced by an arbitrary, but positive definite, position dependent[9] matrix $\boldsymbol{D}_d(\boldsymbol{x})$.

In [ASOFH03] two different methods, how to choose the matrix $\boldsymbol{D}_d(\boldsymbol{x})$ for a given symmetric and positive definite stiffness matrix $\boldsymbol{K}_d$, were proposed and evaluated for the DLR lightweight robots. In the following only the method based on the generalized eigenvalue decomposition of symmetric matrices will be presented in more detail. For this, the following lemma is formulated[10] [Har97].

Lemma 3.6. *Given a symmetric and positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and a symmetric matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$. Then one can find a non-singular matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\boldsymbol{B}_0 \in \mathbb{R}^{n\times n}$, such that $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{Q}^T\boldsymbol{B}_0\boldsymbol{Q}$.*

The elements of this diagonal matrix $\boldsymbol{B}_0$ are called the *generalized eigenvalues* of $\boldsymbol{B}$ with respect to $\boldsymbol{A}$. If the matrix $\boldsymbol{B}$ is positive definite, the generalized eigenvalues will be positive. In order to design the matrix $\boldsymbol{D}_d(\boldsymbol{x})$ in (3.16) a *quasi-static* analysis is performed, which means that in each position $\boldsymbol{x}_0$ the system is approximated by the following linear time-invariant system

$$\boldsymbol{\Lambda}(\boldsymbol{x}_0)\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d(\boldsymbol{x}_0)\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext} \ , \tag{3.19}$$

wherein the Coriolis/centrifugal matrix is neglected and the matrices $\boldsymbol{\Lambda}(\boldsymbol{x}_0)$ and $\boldsymbol{D}_d(\boldsymbol{x}_0)$ are considered as constant. By applying Lemma 3.6 (with $\boldsymbol{\Lambda}(\boldsymbol{x}_0)$ and $\boldsymbol{K}_d$ corresponding to $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively) the matrices $\boldsymbol{K}_d$ and $\boldsymbol{\Lambda}(\boldsymbol{x}_0)$ can be diagonalized simultaneously by a non-singular matrix $\boldsymbol{Q}(\boldsymbol{x}_0)$, such that the system (3.19) can be written in the form

$$\boldsymbol{Q}(\boldsymbol{x}_0)^T\boldsymbol{Q}(\boldsymbol{x}_0)\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d(\boldsymbol{x}_0)\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{Q}(\boldsymbol{x}_0)^T\boldsymbol{B}_0(\boldsymbol{x}_0)\boldsymbol{Q}(\boldsymbol{x}_0)\tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext} \ , \tag{3.20}$$

with a positive definite diagonal matrix $\boldsymbol{B}_0(\boldsymbol{x}_0)$.

Let $\lambda_{K,i}^{\Lambda}$ be the i^{th} diagonal element of $\boldsymbol{B}_0$. This is the i^{th} generalized eigenvalue of $\boldsymbol{K}_d$ with respect to $\boldsymbol{\Lambda}(\boldsymbol{x}_0)$. Then the matrix $\boldsymbol{D}_d(\boldsymbol{x}_0)$ can be chosen as $\boldsymbol{D}_d(\boldsymbol{x}_0) = 2\boldsymbol{Q}(\boldsymbol{x}_0)^T \operatorname{diag}(\xi_i\sqrt{\lambda_{K,i}^{\Lambda}})\boldsymbol{Q}(\boldsymbol{x}_0)$, where ξ_i is a damping factor[11] to be chosen in the range $[0,1]$.

Clearly, by this choice the system can be written in the state variable $\boldsymbol{z} = \boldsymbol{Q}(\boldsymbol{x}_0)\tilde{\boldsymbol{x}}$ in the following form

$$\ddot{\boldsymbol{z}} + 2\operatorname{diag}(\xi_i\sqrt{\lambda_{K,i}^{\Lambda}})\dot{\boldsymbol{z}} + \operatorname{diag}(\lambda_{K,i}^{\Lambda})\boldsymbol{z} = \boldsymbol{Q}(\boldsymbol{x}_0)^{-T}\boldsymbol{F}_{ext} \ . \tag{3.21}$$

It should be mentioned again that the above approximations are only used for the design of the damping matrix and do not affect the stability properties of the system, because it is ensured that the damping matrix is always positive definite.

[9] It can even be chosen time-varying.

[10] The author would like to thank Udo Frese for pointing out this feature of positive definite matrices.

[11] Corresponding to the i^{th} generalized eigenvector.

3.4 Singularity Treatment

In the previous sections it was assumed that $\boldsymbol{J}(\boldsymbol{q})$ is non-singular in the considered workspace. Generally, two different types of singularities are to be distinguished: representation singularities and kinematic singularities. The first ones are the result of the particular choice of local coordinates $\boldsymbol{x}$ which describe the configuration $\boldsymbol{h}_{st}(\boldsymbol{q}) \in SE(3)$ of the tool-frame. It is well known that every minimal representation of $SO(3)$, and hence also of $SE(3)$, will have such singularities. They can in practice be handled by using different representations for different regions of the workspace. Switching between these different representations, however, is a critical issue. The only way to avoid such singularities completely is to use a non-minimal parametrization instead. A method how to construct a singularity-free Cartesian stiffness based on such a non-minimal parametrization will be reviewed in Section 3.5.2.

The second type of singularities, the kinematic singularities, however, cannot be avoided. They are inherently connected to the kinematic structure of the manipulator, representing the inability to produce an arbitrary movement of the end-effector at certain joint configurations. From a mathematical point of view the kinematic singularities are the singularities of the body Jacobian. It is clear that near a singular configuration the desired impedance behavior (3.16) in general cannot be achieved exactly but will be *distorted*. By using a control law like (3.18) the robot simply might *get stuck* when moving through a singular configuration because the second term in (3.18) might vanish.

One method for the control of manipulators at kinematic singularities was proposed in [CK95]. In this work a factorization of the determinant of the body Jacobian is required, in which each term corresponds to one of the different singular configurations of the robot. When approaching such a singular configuration the controller splits up into two parts. One part in the controller is used to control the *distance* of the joint configuration from the next singular configuration via the relevant term in the factorization. The other part basically controls the movement *orthogonally*[12] to the singular direction by the classical operational space formulation [Kha87]. While this is a quite effective approach, especially (but not only) for the use in a position controller, in case of an impedance controller one sometimes[13] prefers simply to avoid the singular configurations.

In this section a possible solution for avoiding the singularities is introduced. This will implicitly restrict the manipulators workspace, but this restriction will be done in a generic way without the need to know all the singular configurations in advance.

In order to implement the singularity avoidance, the Cartesian impedance controller from (3.18) is combined with a second impedance controller which forces the manipulator to move away from singular configurations. The combination of the two controllers can be done by considering the superposition

[12] Notice that the term *orthogonality*, as it was used in [CK95], actually requires the choice of a particular metric.

[13] It clearly depends on the application whether the avoidance of singularities is admissible or not.

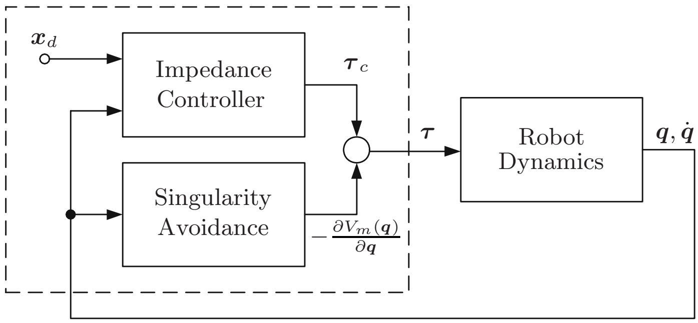

Fig. 3.1. Superposition of the Cartesian impedance controller and the singularity avoidance

principle for impedances [Hog85a, BS98]. According to this multiple impedance components coupled to an admittance may be assembled simply by adding their output torques even if the behavior of the impedances is nonlinear.

For the construction of a singularity avoidance potential, the kinematic manipulability measure [Yos90]

$$m_{kin}(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)} \tag{3.22}$$

is used. The singularity avoidance potential is chosen as

$$V_m(\boldsymbol{q}) = \begin{cases} k_s(m_{kin}(\boldsymbol{q}) - m_0)^2 & m_{kin}(\boldsymbol{q}) \le m_0 \\ 0 & m_{kin}(\boldsymbol{q}) > m_0 \end{cases} , \tag{3.23}$$

where $k_s \in \mathbb{R}$ is a positive scalar factor controlling the gain of the singularity avoidance. The upper bound $m_0 > 0 \in \mathbb{R}$ for $m_{kin}(\boldsymbol{q})$ determines the area around a singular configuration, in which the singularity avoidance will be active. These two parameters are the design parameters.

Let $\boldsymbol{\tau}_c$ denote the output torque of the Cartesian impedance controller from the last section, then the complete control law with singularity avoidance is given by

$$\boldsymbol{\tau} = \boldsymbol{\tau}_c - \frac{\partial V_m(\boldsymbol{q})}{\partial \boldsymbol{q}} . \tag{3.24}$$

Notice that instead of $m_{kin}(\boldsymbol{q})$ one could also use the dynamic manipulability measure $m_{dyn}(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{M}(\boldsymbol{q})^{-T}\boldsymbol{J}(\boldsymbol{q})^T)}$ [Yos90]. With regard to the implementation the use of the kinematic measure has the advantage of consuming less computation power.

Some additional remarks concerning the effects of the singularity avoidance on the stability properties of the closed loop system are in order. Notice that the singularity avoidance will only be active when the manipulator configuration is near to a singularity. Far[14] from the singularities, only the Cartesian impedance

[14] Measured by the condition $m_{kin}(\boldsymbol{q}) > m_0$.

controller is active. If one assumes that the virtual equilibrium position is far away from the singularities and that the desired stiffness is not *too low*, it is clear that the singularity avoidance will not affect the (local) stability of the system. On the other hand it should be mentioned that it indeed could happen that, near to a singularity, the torques due to the Cartesian stiffness potential and the singularity avoidance potential counterbalance each other. Thereby, the complete potential function may have a local minimum other than the desired equilibrium position. Then the robot clearly could get stuck while moving through this local minimum, which, however, will be close to the singular configuration.

3.5 Remarks on the Stiffness Implementation

In the previous sections, it was assumed that the end-effector pose can be described by some end-effector coordinates $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$ and that the Cartesian error $\tilde{\boldsymbol{x}}$ reads accordingly as $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d$. In contrast to the damping matrix discussed in Section 3.3, the desired stiffness matrix $\boldsymbol{K}_d$ is required to be a constant matrix. Otherwise the stability statement from Proposition 3.4 would not be valid any more.

In this section it will be explained how the particular choice of the used Cartesian coordinates $\boldsymbol{f}(\boldsymbol{q})$ in the control law (3.18) affects the resulting stiffness behavior. For this it is assumed that the Cartesian coordinates $\boldsymbol{x} \in \mathbb{R}^6$ can be split up into two components $\boldsymbol{x}_t \in \mathbb{R}^3$ and $\boldsymbol{x}_r \in \mathbb{R}^3$ describing the end-effector position and orientation, respectively. Analogously, the translational error is denoted by $\tilde{\boldsymbol{x}}_t \in \mathbb{R}^3$ and the rotational error by $\tilde{\boldsymbol{x}}_r \in \mathbb{R}^3$. In accordance with the translational and rotational error, the stiffness matrix $\boldsymbol{K}_d$ is partitioned into a translational stiffness $\boldsymbol{K}_t$, a rotational stiffness $\boldsymbol{K}_r$, and a coupling stiffness $\boldsymbol{K}_c$, i.e.

$$\boldsymbol{K}_d = \begin{bmatrix} \boldsymbol{K}_t & \boldsymbol{K}_c \\ \boldsymbol{K}_c^T & \boldsymbol{K}_r \end{bmatrix} .$$

As described in Section 2.1 the pose of the end-effector, i.e. the pose of the tool frame $\mathcal{T}$, is given by the forward kinematics map $\boldsymbol{h}_{st}(\boldsymbol{q}) : \mathcal{Q} \to SE(3)$, which can be computed by the product of exponentials formula (2.3). According to the notation of Chapter 2 the rotational part of $\boldsymbol{h}_{st}(\boldsymbol{q})$ is denoted by $\boldsymbol{R}_{st}(\boldsymbol{q}) \in SO(3)$ and the translational part by $\boldsymbol{p}_{st}(\boldsymbol{q}) \in \mathbb{R}^3$. Given a (time-varying) *virtual* pose $\boldsymbol{h}_{sd}(t) \in SE(3)$, corresponding to a (time-varying) desired frame $\mathcal{D}$, the deviation of $\boldsymbol{h}_{st}(\boldsymbol{q})$ from $\boldsymbol{h}_{sd}(t)$ can be described by

$$\boldsymbol{h}_{dt}(\boldsymbol{q}, t) = \boldsymbol{h}_{sd}^{-1}(t)\boldsymbol{h}_{st}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R}_{dt}(\boldsymbol{q}, t) & {}_d\boldsymbol{p}_{dt}(\boldsymbol{q}, t) \\ \boldsymbol{0} & 1 \end{bmatrix} . \tag{3.25}$$

The following discussion is split up into two parts. Section 3.5.1 clarifies some aspects concerning the implementation of the translational stiffness $\boldsymbol{K}_t$, and

Section 3.5.2 describes different orientation representations according to the rotational stiffness $\boldsymbol{K}_r$. The coupling stiffness $\boldsymbol{K}_c$ can be designed similarly. Since it is not used in the applications of Chapter 9 it is not considered here in detail, i.e. $\boldsymbol{K}_c = \boldsymbol{0}$.

3.5.1 Translational Stiffness

The vector ${}_d\boldsymbol{p}_{dt}(\boldsymbol{q}, t)$ in (3.25) corresponds to a vector from the origin of the desired frame $\mathcal{D}$ to the origin of the tool frame $\mathcal{T}$, represented in frame $\mathcal{D}$. From this one can get an appropriate set of translational end-effector coordinates by rotating this vector into the base frame $\mathcal{S}$

$$\boldsymbol{p}_{dt}(\boldsymbol{q}, t) = \boldsymbol{R}_{sd}(t)\,{}_d\boldsymbol{p}_{dt}(\boldsymbol{q}, t) = \boldsymbol{p}_{st}(\boldsymbol{q}) - \boldsymbol{p}_{sd}(t)\ . \qquad (3.26)$$

Then, one can choose $\boldsymbol{x}_t = \boldsymbol{p}_{st}(\boldsymbol{q})$ as translational coordinates and $\boldsymbol{p}_{sd}(t)$ as their corresponding virtual equilibrium position. By this choice the translational part of the stiffness matrix corresponds to a stiffness, which is represented in the base frame $\mathcal{S}$, and consequently the eigenvectors of $\boldsymbol{K}_t$, corresponding to the principal axes of the stiffness matrix, are constant vectors expressed in the base frame.

For some applications it is instead desired to use a stiffness implementation in which the principal axes of the stiffness matrix are defined (as constant vectors) in the desired frame $\mathcal{D}$ or the tool frame $\mathcal{T}$. Then, one must refer to a different set of coordinates. If for instance the vector ${}_d\boldsymbol{p}_{dt}(\boldsymbol{q}, t)$ is used as the translational part $\tilde{\boldsymbol{x}}_t$ of the Cartesian error $\tilde{\boldsymbol{x}}$, this corresponds to a stiffness representation in $\mathcal{D}$. For a stiffness representation in $\mathcal{T}$ one can accordingly use translational coordinates of the form ${}_t\boldsymbol{p}_{dt}(\boldsymbol{q}, t) = \boldsymbol{R}_{td}(\boldsymbol{q}, t)\,{}_d\boldsymbol{p}_{dt}(\boldsymbol{q}, t)$. Notice that (except for the choice $\boldsymbol{x}_t = \boldsymbol{p}_{st}(\boldsymbol{q})$) one cannot write $\tilde{\boldsymbol{x}}_t$ as the difference between a configuration dependent coordinate $\boldsymbol{x}_t(\boldsymbol{q})$ and a time-varying (but configuration-independent) virtual equilibrium position $\boldsymbol{x}_d(t)$ any more. However, the impedance controllers of the last sections can of course be easily adapted to this case. This will be treated in more detail in Section 3.5.3.

3.5.2 Rotational Stiffness

While the procedure for the translational part of the stiffness was quite straightforward, the situation is more complex for the orientation representation. It is well known that no global minimal representation of $SO(3)$ exists. Different choices of orientation coordinates for the use in Cartesian controllers were analyzed in detail in [Nat03]. In this section two different approaches for the implementation of a rotational stiffness are shortly discussed, namely *Euler angles* and *unit quaternions*.

Consider a set of Euler angles $\boldsymbol{\phi}_{st}(\boldsymbol{q}) = \boldsymbol{\phi}(\boldsymbol{R}_{st}(\boldsymbol{q})) \in \mathbb{R}^3$ computed directly from the rotation matrix $\boldsymbol{R}_{st}(\boldsymbol{q})$. One can build the *classical* Euler angle based orientation error $\boldsymbol{\phi}^*_{dt}(\boldsymbol{q}, t) = \boldsymbol{\phi}_{st}(\boldsymbol{q}) - \boldsymbol{\phi}_{sd}(t)$. But this approach to formulate the orientation error has a major drawback: Singularities are encountered whenever either the robot orientation $\boldsymbol{\phi}_{st}(\boldsymbol{q})$ or the desired equilibrium orientation $\boldsymbol{\phi}_{sd}(t)$ is singular. This can happen for arbitrary small orientation errors. A much better approach is to compute the Euler angles from the difference frame $\boldsymbol{h}_{td}(\boldsymbol{q},t)$ via ${}_t\boldsymbol{\phi}_{td}(\boldsymbol{q},t) = \boldsymbol{\phi}(\boldsymbol{R}_{td}(\boldsymbol{q},t))$. This representation, sometimes called *modified Euler angle representation*, is much more robust against singularities. If, for instance, the well known *roll-pitch-yaw* representation is used, then the representation does not contain any singularities up to an orientation error of $\pi/2$.

The main disadvantage of an Euler angle based stiffness is the occurrence of representation singularities. These can only be avoided if one refers to a non-minimal representation of the end-effector orientation. Herein, only the use of the so-called *unit quaternions*[15] will be shortly discussed. A detailed exposition of unit quaternions is beyond the scope of this section, only some relevant properties are reported. More details on the use of unit quaternions can be found e.g. in [CNSV99, Nat03] and the references cited therein.

A unit quaternion $(\eta, \boldsymbol{\epsilon})$ consists of a scalar part $\eta \in \mathbb{R}$ and a vector part $\boldsymbol{\epsilon} \in \mathbb{R}^3$, which fulfill the condition $\eta^2 + \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = 1$. The relation between a rotation matrix and unit quaternions is given as follows. Suppose that a rotation matrix is specified by a rotation of an angle α about an axis $\boldsymbol{r}$ (with $||\boldsymbol{r}||_2 = 1$), then the corresponding unit quaternion is given by $\eta = \cos(\alpha/2)$, $\boldsymbol{\epsilon} = \boldsymbol{r}\sin(\alpha/2)$. With the restriction of α to the interval $[-\pi, \pi]$ the set of unit quaternions is a one-to-one covering of $SO(3)$. In contrast to Euler angles, unit quaternions give a singularity-free representation of $SO(3)$.

Let $\boldsymbol{\epsilon}_{dt}(\boldsymbol{q},t)$ be the vector part of the unit quaternion according to the rotation matrix $\boldsymbol{R}_{dt}(\boldsymbol{q},t)$. For the implementation of a rotational stiffness, one can then refer to the following orientation error $\tilde{\boldsymbol{x}}_r(\boldsymbol{q},t) := 2\boldsymbol{\epsilon}_{dt}(\boldsymbol{q},t)$. Notice that this quantity (and thus also the resulting stiffness term) is periodic with respect to α.

Herein, only the most commonly used types of stiffness implementations were discussed. Their relation to the controllers from this book will be clarified in the next section. For another interesting class of stiffness implementations the reader is referred to [FH95, FB97, Fas97, ZF00, SD01, Str01], in which the so-called *spatial* stiffness is discussed.

3.5.3 Consequences for the Closed Loop Dynamics

The previous two sections led to a Cartesian error $\tilde{\boldsymbol{x}}(\boldsymbol{q},t)$ which cannot be written as the difference between an actual and a desired quantity. In the following it is shown how the model from Section 3.1.2 must be adapted in order to cope with this situation.

The first and second derivatives of the error vector $\tilde{\boldsymbol{x}}(\boldsymbol{q},t)$ are given by

$$\dot{\tilde{\boldsymbol{x}}} = \frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial t} \ ,$$

$$\ddot{\tilde{\boldsymbol{x}}} = \frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}\ddot{\boldsymbol{q}} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}\right)\dot{\boldsymbol{q}} + \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial t} \ .$$

[15] For unit quaternions also the term *Euler parameters* is sometimes used.

These equations replace (3.2) and (3.3). The equations of motion for the error vector $\tilde{\boldsymbol{x}}$ can now be derived by following the same steps as in Section 3.1.2. For the ease of presentation the following substitutions are made

$$\frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial \boldsymbol{q}} = \boldsymbol{J}_x(\boldsymbol{q},t) \ ,$$
$$\frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{q},t)}{\partial t} = -\boldsymbol{v}_t(\boldsymbol{q},t) \ .$$

Therefore, one gets the following equations of motion

$$\boldsymbol{\Lambda}(\boldsymbol{q},t)(\ddot{\tilde{\boldsymbol{x}}} + \dot{\boldsymbol{v}}_t(\boldsymbol{q},t)) + \boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}},t)(\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{v}_t(\boldsymbol{q},t)) + \boldsymbol{J}_x(\boldsymbol{q},t)^{-T}\boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_x(\boldsymbol{q},t)^{-T}(\boldsymbol{\tau} + \boldsymbol{\tau}_{ext}) = \boldsymbol{0} \ ,$$

where $\boldsymbol{\Lambda}(\boldsymbol{q},t)$ and $\boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}},t)$ are the relevant Cartesian inertia matrix and Coriolis/centrifugal matrix

$$\boldsymbol{\Lambda}(\boldsymbol{q},t) = \boldsymbol{J}_x(\boldsymbol{q},t)^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}_x(\boldsymbol{q},t)^{-1} \ ,$$
$$\boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}},t) = \boldsymbol{J}_x(\boldsymbol{q},t)^{-T}\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}_x(\boldsymbol{q},t)^{-1}\dot{\boldsymbol{J}}_x(\boldsymbol{q},t)\right)\boldsymbol{J}_x(\boldsymbol{q},t)^{-1} \ ,$$

analogous to (3.10) and (3.11). Comparing this equation with (3.12), one can see that the same controllers as in the previous sections can be used, when the following substitutions are made

$$\boldsymbol{J}(\boldsymbol{q}) \rightarrow \boldsymbol{J}_x(\boldsymbol{q},t) \ ,$$
$$\dot{\boldsymbol{x}}_d \rightarrow -\boldsymbol{v}_t(\boldsymbol{q},t) \ ,$$
$$\ddot{\boldsymbol{x}}_d \rightarrow -\dot{\boldsymbol{v}}_t(\boldsymbol{q},t) \ .$$

Notice that, although these quantities become time-varying now, these substitutions do not alter the stability statements presented in this book.

Explicit formulas of $\boldsymbol{J}_x(\boldsymbol{q},t)$ and $\boldsymbol{v}_t(\boldsymbol{q},t)$ for the different sets of local coordinates can be given in terms of the body Jacobian $\boldsymbol{J}^b(\boldsymbol{q})$. Notice therefore that $\boldsymbol{J}^b(\boldsymbol{q})$, as defined in Section 2.1, is used to compute the translational velocity ${}_t\boldsymbol{v}_{st} = \boldsymbol{R}_{ts}\dot{\boldsymbol{p}}_{st}$ and the angular velocity ${}_t\boldsymbol{\omega}_{st} = \boldsymbol{R}_{ts}\boldsymbol{\omega}_{st}$, expressed in the tool frame

$$\begin{pmatrix} {}_t\boldsymbol{v}_{st} \\ {}_t\boldsymbol{\omega}_{st} \end{pmatrix} = \boldsymbol{J}^b(\boldsymbol{q})\dot{\boldsymbol{q}} \ . \tag{3.27}$$

3.6 Summary

In this chapter the Cartesian impedance control problem for a (conventional) robot model without joint flexibility was discussed. First the general solution according to a desired impedance in form of a generalized mass-spring-damper system was treated. Then the situation for an impedance controller without inertia shaping was analyzed in detail. In particular this brought up the problem of designing the Cartesian damping matrix appropriately. For this problem a solution based on the simultaneous diagonalization of positive definite matrices was proposed. Moreover, the singularity avoidance problem was discussed, and a solution based on the superposition of impedances [Hog85c] was proposed. Then some additional details on the choice of Cartesian coordinates were discussed which lead to stiffness representations with respect to different frames. These issues, though practically relevant, are only rarely addressed in the robotics literature. Finally, an alternative orientation stiffness implementation which avoids the analytical singularities inherent to any minimal representation of $SO(3)$ was briefly discussed for further reference.