29. Cooperative Manipulators

Fabrizio Caccavale, Masaru Uchiyama

This chapter is devoted to cooperative manipulation of a common object by means of two or more robotic arms. The chapter opens with a historical overview of the research on cooperative manipulation, ranging from early 1970s to very recent years. Kinematics and dynamics of robotic arms cooperatively manipulating a tightly grasped rigid object are presented in depth. As for the kinematics and statics, the chosen approach is based on the so-called symmetric formulation; fundamentals of dynamics and reduced-order models for closed kinematic chains are discussed as well. A few special topics, such as the definition of geometrically meaningful cooperative task space variables, the problem of load distribution, and the definition of manipulability ellipsoids, are included to give the reader a complete picture of modeling and evaluation methodologies for cooperative manipulators. Then, the chapter presents the main strategies for controlling both the motion of the cooperative system and the interaction forces between the manipulators and the grasped object; in detail, fundamentals of hybrid force/position control, proportional–derivative (PD)-type force/position schemes, feedback linearization techniques, and impedance control approaches are given. In the last section further reading on advanced topics related to control of cooperative robots is suggested; in detail, advanced nonlinear control strategies are briefly discussed (i. e., intelligent control approaches, synchronization control, decentralized control); also, fundamental results on modeling and control of cooperative systems possessing some degree of flexibility are briefly outlined.

29.1 A Historical Overview

It was not long after the emergence of robotics technologies that multi-arm robot systems began to be investigated by robotics researchers. In the early 1970s, research on this topic had already started. This interest was mainly due to typical limitations in applications of single-arm robots. It has been recognized, in fact, that many tasks that are difficult or impossible to execute by a single robot become feasible when two or more manipulators are employed in a cooperative way. Such tasks include, for instance, carrying heavy or large payloads, the assembly of multiple parts without using special fixtures, and handling of objects that are flexible or possess extra degrees of freedom. Research on this subject has been aimed at solving existing problems and opening up a new stream of applications in flexible manufacturing systems as well as in poorly structured environments (e.g., outer space and undersea).

Examples of research work in the early days include that by *Fujii* and *Kurono* [29.1], *Nakano* et al. [29.2], and *Takase* et al. [29.3]. Already in those pieces of work important key issues in the control of multi-arm robots were investigated: master/slave control, force/compliance control, and task-space control. In [29.1] *Fujii* and *Kurono* proposed to adopt a compliance control concept for the coordination of multiple manipulators; they defined a task vector with respect to the object frame and controlled the compliance expressed in the same coordinate frame. An interesting feature of the work by *Fujii* and *Kurono* [29.1] and by *Takase* et al. [29.3] is that force/compliance control was implemented by exploiting back-drivability of the actuators, without using force/torque sensors. The importance of this technique in practical applications, however, was not recognized at that time, and more complex approaches using force/torque sensors lured people in robotics. *Nakano* et al. [29.2, 4] proposed a master/slave force control approach for the coordination of two arms carrying an object cooperatively and pointed out the necessity of force control for cooperative multiple robots.

In the 1980s, based on several fundamental theoretical results for single-arm robots, strong research on multi-arm robotic systems was renewed [29.5]. Definition of task vectors with respect to the object to be handled [29.6], dynamics and control of the closed kinematic chain formed by the multi-arm robot and the object [29.7, 8], and force control issues, such as hybrid position/force control [29.9–12], were explored. Through this research work, a strong theoretical background for the control of multi-arm robots has been formed, providing the basis for research on more advanced topics from the 1990s to today.

How to parameterize the constraint forces/moments on the object based on the dynamic model of the whole cooperative system has been recognized as a critical issue; in fact, this parametrization leads to the definition of task variables for the control and hence to an answer to one of the most frequently asked questions in the field of multi-arm robotics: how to control simultaneously the trajectory of the object, the mechanical stresses (internal forces/moments) acting on the object, load sharing among the arms, and even the external forces/moments on the object. Force decomposition may be a key to solving these problems and has been studied by *Uchiyama* and *Dauchez* [29.11, 12] and *Walker* et al. [29.13] as well as *Bonitz* and *Hsia* [29.14]. In detail, devising a geometrically clear parametrization of the internal forces/moments acting on the object has been recognized as an important problem; *Williams* and *Khatib* have given a solution to this [29.15, 16]. Several cooperative control schemes based on the sought parameterizations have been designed, including control of motion and force [29.11, 12, 17–19] and impedance/compliance control [29.20–22]. Other approaches include adaptive control [29.23, 24], kinematic control [29.25], task-space regulation [29.26], joint-space control [29.27, 28], and coordinated control [29.29].

Also, the definition of user-oriented task-space variables for coordinated control [29.26] and the development of meaningful performance measures [29.30–33] have been fruitfully investigated in 1990s.

Load sharing among the arms is also an interesting issue on which many papers have been published [29.34–39]. Load sharing may be exploited both for optimal load distribution among arms and for robust holding of the object, when the object is held by the arms without being grasped rigidly. In both cases, anyhow, this becomes a problem of optimization and can be solved by either heuristic [29.40] or mathematical methods [29.41].

Other research efforts have been focused on cooperative handling of multibodied or even flexible objects [29.42–44]. Control of multi-flexible-arm robots has been investigated [29.45, 46], since the merits of flexible-arm robots can be exploited in cooperative systems [29.47], i. e., lightweight structure, intrinsic compliance and hence safety, etc.

Robust holding of an object in the presence of slippage of end-effectors on the object may be achieved as well, if the slippage is detected correctly [29.48].

A more recent control framework for cooperative systems is so-called synchronization control [29.49, 50]; in this class of approaches the control problem is formulated in terms of suitably defined errors accounting for the motion synchronization between the manipulators involved in the cooperative task. As for the nonlinear control of cooperative manipulation systems, efforts have been spent on intelligent control (see, e.g., [29.51] and [29.52], where fuzzy control is exploited to cope with unmodeled dynamics, parametric uncertainties, and disturbances) as well as on the investigation of control strategies in the presence of partial state feedback [29.53].

The problem of the implementation of cooperative control strategies on conventional industrial robots has attracted the increasing interest of the research community. In fact, the control units of industrial robots do not present all the features needed to implement nonlinear torque control schemes, while the integration of force/torque sensing in standard industrial robot control units is often cumbersome and tends to be avoided in industry for many reasons: unreliability, cost, etc. Hence, the rebirth of the early methods, where force sensors were not used (*Fujii* and *Kurono* [29.1], *Inoue* [29.54]), has become attractive for industrial settings. Hybrid position/force control without using force/torque sensors has been successfully implemented [29.55]. Interesting results on the implementation of effective cooperative control strategies for industrial robots are presented, e.g., in [29.56], where a design approach that makes use of tool-based coordinate systems, trajectory generation, and distributed control of multiple robots is presented.

Another interesting aspect, related to the reliability and safety of cooperative manipulation systems, is investigated in [29.57], where the use of nonrigid grippers is considered to avoid large internal forces and, at the same time, to achieve safe manipulation of the object, even in the presence of failures and unpredicted contacts with the external environment.

Finally, it is worth mentioning the strict relationship between problems related to grasping of objects by fingers/hands (widely described in Chap. 28) and those related to cooperative manipulation. In fact, in both cases, multiple manipulation structures grasp a commonly manipulated object. In multifingered hands only some motion components are transmitted through the contact point to the manipulated object (unilateral constraints), while cooperative manipulation via robotic arms is achieved by rigid (or near-rigid) grasp points and interaction takes place by transmitting all the motion components through the grasping points (bilateral constraints). While many common problems between the two fields can be tackled in a conceptually similar way (e.g., kinetostatic modeling, force control), many other are specific of each of the two application fields (e.g., form and force closure for multifingered hands). Interestingly, a common frame for both cooperative and multifingered manipulation has been proposed [29.25]; namely rolling/sliding of the grasp points on the object is taken into account by modeling the contacts via rotational/prismatic joints; then, the desired manipulator/finger joints trajectories are derived from the desired object motion by using numerical inverse kinematics algorithms.

29.2 Kinematics and Statics

Consider a system composed by M manipulators, each equipped with N_i joints ($i = 1, \dots, M$). Let $\boldsymbol{p}_i$ denote the (3×1) vector of the position of the i-th end-effector coordinate frame, $\mathcal{T}_i$, with respect to a common base frame, $\mathcal{T}$; let $\boldsymbol{R}_i$ denote the (3×3) matrix which expresses the orientation of $\mathcal{T}_i$ with respect to the base frame $\mathcal{T}$.

Both $\boldsymbol{p}_i$ and $\boldsymbol{R}_i$ can be expressed as a function of the $(N_i \times 1)$ vector of the joints variables of each manipulator $\boldsymbol{q}_i$ through the direct kinematics equations

$$\begin{cases} \boldsymbol{p}_i = \boldsymbol{p}_i(\boldsymbol{q}_i)\,, \\ \boldsymbol{R}_i = \boldsymbol{R}_i(\boldsymbol{q}_i)\,. \end{cases} \tag{29.1}$$

Of course, the orientation of the end-effector frame may be expresses in terms of a minimal set of angles, i. e., a set of three Euler angles $\boldsymbol{\phi}_i$. In this case, direct kinematics can be expressed in terms of an operational space vector $\boldsymbol{x}_i$ as follows

$$\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{q}_i) = \begin{pmatrix} \boldsymbol{p}_i(\boldsymbol{q}_i) \\ \boldsymbol{\phi}_i(\boldsymbol{q}_i) \end{pmatrix}\,. \tag{29.2}$$

The linear $\dot{\boldsymbol{p}}_i$ and angular $\boldsymbol{\omega}_i$ velocity of the i-th end-effector can be collected in the (6×1) generalized velocities vector $\boldsymbol{v}_i = \left(\dot{\boldsymbol{p}}_i^\top\ \boldsymbol{\omega}_i^\top\right)^\top$. Then, the differential direct kinematics can be expressed as

$$\boldsymbol{v}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i\,, \tag{29.3}$$

where the $(6\times N_i)$ matrix $\boldsymbol{J}_i$ is the so-called *geometric Jacobian* of the i-th manipulator (see Chap. 1). When the velocity is expressed as the derivative of the operational space vector, differential kinematics takes a formally similar form

$$\dot{\boldsymbol{x}}_i = \frac{\partial \boldsymbol{f}_i(\boldsymbol{q}_i)}{\partial \boldsymbol{q}_i}\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{\mathrm{A}i}(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i\,, \tag{29.4}$$

where the $(6\times N_i)$ matrix $\boldsymbol{J}_{\mathrm{A}i}$ is the so-called *analytical Jacobian* of the i-th manipulator (see Chap. 1).

Let us consider the (6×1) vector of the generalized forces acting at the i-th end-effector

$$\boldsymbol{h}_i = \begin{pmatrix} \boldsymbol{f}_i \\ \boldsymbol{n}_i \end{pmatrix}\,, \tag{29.5}$$

where $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ denote the force and moment, respectively. By invoking the principle of virtual work, a relation dual to (29.3) can be derived

$$\boldsymbol{\tau}_i = \boldsymbol{J}^\top(\boldsymbol{q}_i)\boldsymbol{h}_i \ , \tag{29.6}$$

where $\boldsymbol{\tau}_i$ is the $(N_i \times 1)$ vector of the forces/torques acting at the joints of the i-th manipulator.

For the sake of simplicity, consider a system of two cooperative robots (Fig. 29.1) manipulating a common object. Let C be a fixed point of the object (e.g., the center of mass of the object), whose position in the base frame is given by the vector $\boldsymbol{p}_\mathrm{C}$; moreover, let $\mathcal{T}_\mathrm{C}$ be a coordinate frame attached to object with origin in C. The *virtual stick* [29.11, 12] is defined as the vector $\boldsymbol{r}_i$ $(i = 1, 2)$ which determines the position of $\mathcal{T}_\mathrm{C}$ with respect to $\mathcal{T}_i$ $(i = 1, 2)$; it is assumed that $\boldsymbol{r}_i$ behaves as a rigid stick fixed to the i-th end-effector. Hence, each virtual stick, expressed in the frame $\mathcal{T}_i$ (or $\mathcal{T}_\mathrm{C}$), is a constant vector if the commonly grasped object is rigid and tightly grasped by each manipulator. In this case, the direct kinematics of each manipulator can be expressed in terms of a virtual end-effector frame $\mathcal{T}_{\mathrm{S},i} = \mathcal{T}_\mathrm{C}$, having the same orientation of $\mathcal{T}_\mathrm{C}$ and origin located in $\boldsymbol{p}_{\mathrm{S},i} = \boldsymbol{p}_i + \boldsymbol{r}_i = \boldsymbol{p}_\mathrm{C}$. Hence, the position and orientation at the tip of each virtual stick are given by $(i = 1, 2)$

$$\boldsymbol{p}_{\mathrm{S},i} = \boldsymbol{p}_\mathrm{C} \ , \quad \boldsymbol{R}_{\mathrm{S},i} = \boldsymbol{R}_\mathrm{C} \ .$$

The set of Euler angles corresponding to $\boldsymbol{R}_{\mathrm{S},i}$ will be denoted by the (3×1) vector $\boldsymbol{\phi}_{\mathrm{S},i}$. Hereafter, it is assumed that the object grasped by the two manipulators can be considered rigid (or nearly rigid) and tightly (or nearly tightly) attached to each end-effector. Hence, the displacements between the above defined frames can be considered null or negligible. Otherwise, if the manipulated object is subject to deformation (e.g., in the case of a flexible object) and/or the grasp is not tight (e.g., in the case of compliant grippers) the above defined coordinate frames, will be subject to nonnegligible displacements with respect to each other.

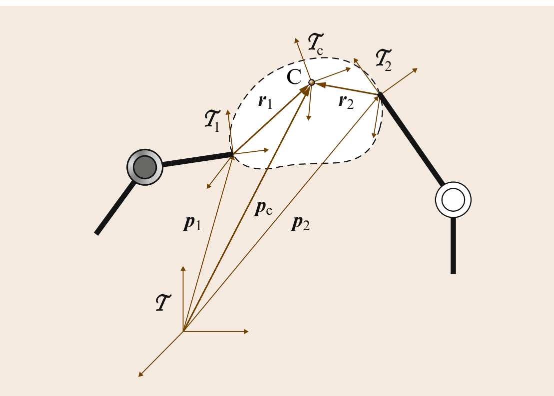

Fig. 29.1 Grasp geometry for a two-manipulator cooperative system manipulating a common object

Let $\boldsymbol{h}_{\mathrm{S},i}$ denote the vector of generalized forces acting at the tip of the i-th virtual stick; it can be easily verified that the following equation holds

$$\boldsymbol{h}_{\mathrm{S},i} = \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\boldsymbol{S}(\boldsymbol{r}_i) & \boldsymbol{I}_3 \end{pmatrix} \boldsymbol{h}_i = \boldsymbol{W}_i \boldsymbol{h}_i \ , \tag{29.7}$$

where $\boldsymbol{O}_l$ and $\boldsymbol{I}_l$ denote, respectively, the null matrix and the identity matrix of $(l \times l)$ dimensions, and $\boldsymbol{S}(\boldsymbol{r}_i)$ is the (3×3) skew-symmetric matrix operator performing the cross product. It is worth noticing that $\boldsymbol{W}_i$ is always full rank.

By invoking the virtual works principle, a relation dual to (29.7) can easily be derived

$$\boldsymbol{v}_i = \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{S}(\boldsymbol{r}_i) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{pmatrix} \boldsymbol{v}_{\mathrm{S},i} = \boldsymbol{W}_i^\top \boldsymbol{v}_{\mathrm{S},i} \ , \tag{29.8}$$

where $\boldsymbol{v}_{\mathrm{S},i}$ is the generalized velocity vector of the virtual stick endpoint. When $\boldsymbol{r}_i = \boldsymbol{0}$, it is $\boldsymbol{W}_i = \boldsymbol{I}_6$; in other words, if the end-effector kinematics of each manipulator is referred to the corresponding virtual stick (or the object reduces to point), the forces and velocities at the two end-effectors coincide with their counterparts referred at the virtual sticks.

29.2.1 Symmetric Formulation

The kinetostatic formulation proposed by *Uchiyama* and *Dauchez* [29.12], i. e., the so-called *symmetric formulation*, is based on kinematic and static relationships between generalized forces/velocities acting at the object and their counterparts acting at the manipulators end-effectors (or at the virtual sticks endpoints).

Let us define first the *external forces* as the (6×1) vector of generalized forces given by

$$\boldsymbol{h}_\mathrm{E} = \boldsymbol{h}_{\mathrm{S},1} + \boldsymbol{h}_{\mathrm{S},2} = \boldsymbol{W}_\mathrm{S}\, \boldsymbol{h}_\mathrm{S} \ , \tag{29.9}$$

with $\boldsymbol{W}_\mathrm{S} = (\boldsymbol{I}_6\ \boldsymbol{I}_6)$ and $\boldsymbol{h}_\mathrm{S} = \left(\boldsymbol{h}_{\mathrm{S},1}^\top\ \boldsymbol{h}_{\mathrm{S},2}^\top\right)^\top$; in other words, $\boldsymbol{h}_\mathrm{E}$ represents the vector of generalized forces causing the object's motion. From (29.7) and (29.9) it follows that $\boldsymbol{h}_\mathrm{E}$ can be expressed in terms of end-effector forces as well

$$\boldsymbol{h}_\mathrm{E} = \boldsymbol{W}_1\, \boldsymbol{h}_1 + \boldsymbol{W}_2\, \boldsymbol{h}_2 = \boldsymbol{W}\boldsymbol{h} \ , \tag{29.10}$$

with $\boldsymbol{W} = (\boldsymbol{W}_1\ \boldsymbol{W}_2)$ and $\boldsymbol{h} = \left(\boldsymbol{h}_1^\top\ \boldsymbol{h}_2^\top\right)^\top$. It can be recognized that $\boldsymbol{W}_\mathrm{S}$ $(\boldsymbol{W})$ is a (6×12) matrix, having a six-dimensional range space and a six-dimensional null space, describing the geometry of the grasp, and thus is usually termed the *grasp matrix*.

The inverse solution to (29.9) yields

$$\boldsymbol{h}_{\mathrm{S}} = \boldsymbol{W}_{\mathrm{S}}^{\dagger}\,\boldsymbol{h}_{\mathrm{E}} + \boldsymbol{V}_{\mathrm{S}}\,\boldsymbol{h}_{\mathrm{I}} = \boldsymbol{U}_{\mathrm{S}}\,\boldsymbol{h}_{\mathrm{O}}\,, \tag{29.11}$$

where $\boldsymbol{W}_{\mathrm{S}}^{\dagger}$ denotes the Moore–Penrose pseudoinverse of $\boldsymbol{W}_{\mathrm{S}}$

$$\boldsymbol{W}_{\mathrm{S}}^{\dagger} = \frac{1}{2}\begin{pmatrix}\boldsymbol{I}_6\\ \boldsymbol{I}_6\end{pmatrix}\,. \tag{29.12}$$

$\boldsymbol{V}_{\mathrm{S}}$ is a matrix whose columns are a basis of the null space of $\boldsymbol{W}_{\mathrm{S}}$, e.g.,

$$\boldsymbol{V}_{\mathrm{S}} = \begin{pmatrix}-\boldsymbol{I}_6\\ \boldsymbol{I}_6\end{pmatrix}\,, \tag{29.13}$$

$\boldsymbol{h}_{\mathrm{O}} = \left(\boldsymbol{h}_{\mathrm{E}}^{\top}\ \boldsymbol{h}_{\mathrm{I}}^{\top}\right)^{\top}$, and

$$\boldsymbol{U}_{\mathrm{S}} = \left(\boldsymbol{W}_{\mathrm{S}}^{\dagger}\ \boldsymbol{V}_{\mathrm{S}}\right)\,. \tag{29.14}$$

The second term on the right-hand side of (29.11), i. e., $\boldsymbol{V}_{\mathrm{S}}\boldsymbol{h}_{\mathrm{I}}$, represents a vector of generalized forces, referred at the tip of the virtual sticks, which lies in the null space of $\boldsymbol{W}_{\mathrm{S}}$; thus, such forces do not contribute to external forces. Hence, the (6×1) vector $\boldsymbol{h}_{\mathrm{I}}$ is a generalized force which does not contribute to the object's motion. Therefore, it represents internal loading of the object (i. e., mechanical stresses) and is termed the *internal forces* vector [29.12]. A similar argument can be used for equation (29.10), whose inverse solution is

$$\boldsymbol{h} = \boldsymbol{W}^{\dagger}\boldsymbol{h}_{\mathrm{E}} + \boldsymbol{V}\boldsymbol{h}_{\mathrm{I}} = \boldsymbol{U}\boldsymbol{h}_{\mathrm{O}}\,, \tag{29.15}$$

where

$$\boldsymbol{U} = (\boldsymbol{W}^{\dagger}\ \boldsymbol{V})\,. \tag{29.16}$$

In [29.13] it has been shown that the first term on the right-hand side of (29.15) represents only contributions to external forces if the pseudoinverse of the grasp matrix is properly defined, i. e.,

$$\boldsymbol{W}^{\dagger} = \begin{pmatrix}\frac{1}{2}\boldsymbol{I}_3 & \boldsymbol{O}_3\\ \frac{1}{2}\boldsymbol{S}(\boldsymbol{r}_1) & \frac{1}{2}\boldsymbol{I}_3\\ \frac{1}{2}\boldsymbol{I}_3 & \boldsymbol{O}_3\\ \frac{1}{2}\boldsymbol{S}(\boldsymbol{r}_2) & \frac{1}{2}\boldsymbol{I}_3\end{pmatrix}\,. \tag{29.17}$$

As $\boldsymbol{V}_{\mathrm{S}}$ in (29.11), the columns of $\boldsymbol{V}$ span the null space of $\boldsymbol{W}$, and can be chosen as [29.30]

$$\boldsymbol{V} = \begin{pmatrix}-\boldsymbol{I}_3 & \boldsymbol{O}_3\\ -\boldsymbol{S}(\boldsymbol{r}_1) & -\boldsymbol{I}_3\\ \boldsymbol{I}_3 & \boldsymbol{O}_3\\ \boldsymbol{S}(\boldsymbol{r}_2) & \boldsymbol{I}_3\end{pmatrix}\,. \tag{29.18}$$

A different parametrization of the inverse solutions to (29.9) and (29.10) can be expressed, respectively, as

$$\boldsymbol{h}_{\mathrm{S}} = \boldsymbol{W}_{\mathrm{S}}^{\dagger}\boldsymbol{h}_{\mathrm{E}} + (\boldsymbol{I}_{12} - \boldsymbol{W}_{\mathrm{S}}^{\dagger}\boldsymbol{W}_{\mathrm{S}})\boldsymbol{h}_{\mathrm{S}}^{\star} \tag{29.19}$$

and

$$\boldsymbol{h} = \boldsymbol{W}^{\dagger}\boldsymbol{h}_{\mathrm{E}} + (\boldsymbol{I}_{12} - \boldsymbol{W}^{\dagger}\boldsymbol{W})\boldsymbol{h}^{\star}\,, \tag{29.20}$$

where $\boldsymbol{h}_{\mathrm{S}}^{\star}$ ($\boldsymbol{h}^{\star}$) is an arbitrary (12×1) vector of generalized forces acting at the tip of the i-th virtual stick (i-th end-effector) projected onto the null space of $\boldsymbol{W}_{\mathrm{S}}$ ($\boldsymbol{W}$) via $\boldsymbol{I}_{12} - \boldsymbol{W}_{\mathrm{S}}^{\dagger}\boldsymbol{W}_{\mathrm{S}}$ ($\boldsymbol{I}_{12} - \boldsymbol{W}^{\dagger}\boldsymbol{W}$).

By exploiting the principle of virtual work, the mappings between generalized velocities, dual to those derived above, can be established. In detail, the mapping dual to (29.11) is:

$$\boldsymbol{v}_{\mathrm{O}} = \boldsymbol{U}_{\mathrm{S}}^{\top}\,\boldsymbol{v}_{\mathrm{S}}\,, \tag{29.21}$$

where $\boldsymbol{v}_{\mathrm{S}} = \left(\boldsymbol{v}_{\mathrm{S},1}^{\top}\ \boldsymbol{v}_{\mathrm{S},2}^{\top}\right)^{\top}$, $\boldsymbol{v}_{\mathrm{O}} = \left(\boldsymbol{v}_{\mathrm{E}}^{\top}\ \boldsymbol{v}_{\mathrm{I}}^{\top}\right)^{\top}$. The vector $\boldsymbol{v}_{\mathrm{E}}$ can be interpreted as the absolute velocity of the object, while $\boldsymbol{v}_{\mathrm{I}}$ represents the relative velocity between the two coordinate frames $\mathcal{T}_{\mathrm{S},1}$ and $\mathcal{T}_{\mathrm{S},2}$ attached to the tips of the virtual sticks [29.12]; this vector is null when the manipulated object is rigid and rigidly grasped. In a similar way, from (29.15), the following mapping can be devised

$$\boldsymbol{v}_{\mathrm{O}} = \boldsymbol{U}^{\top}\,\boldsymbol{v}\,, \tag{29.22}$$

where $\boldsymbol{v} = \left(\boldsymbol{v}_1^{\top}\ \boldsymbol{v}_2^{\top}\right)^{\top}$.

Moreover, a set of position and orientation variables corresponding to $\boldsymbol{v}_{\mathrm{E}}$ and $\boldsymbol{v}_{\mathrm{I}}$ can be defined as follows [29.12, 25]

$$\boldsymbol{p}_{\mathrm{E}} = \frac{1}{2}(\boldsymbol{p}_{\mathrm{S},1} + \boldsymbol{p}_{\mathrm{S},2})\,, \quad \boldsymbol{p}_{\mathrm{I}} = \boldsymbol{p}_{\mathrm{S},2} - \boldsymbol{p}_{\mathrm{S},1}\,, \tag{29.23}$$

$$\boldsymbol{R}_{\mathrm{E}} = \boldsymbol{R}_1\boldsymbol{R}^1(\boldsymbol{k}_{21}^1, \vartheta_{21}/2)\,, \quad \boldsymbol{R}_{\mathrm{I}}^1 = \boldsymbol{R}_2^1\,, \tag{29.24}$$

where $\boldsymbol{R}_2^1 = \boldsymbol{r}_1^{\top}\boldsymbol{R}_2$ is the matrix expressing the orientation of $\mathcal{T}_2$ with respect to the axes of $\mathcal{T}_1$, while $\boldsymbol{k}_{21}^1$ and ϑ_{21} are, respectively, the equivalent unit vector (expressed with respect to $\mathcal{T}_1$) and rotation angle that determine the mutual orientation expressed by $\boldsymbol{R}_2^1$. Hence, $\boldsymbol{R}_{\mathrm{E}}$ expresses a rotation about $\boldsymbol{k}_{21}^1$ by an angle which is half the angle needed to align $\mathcal{T}_2$ with $\mathcal{T}_1$.

In turn, if the orientation variables are expressed in terms of Euler angles, a set of operational space variables can be defined as

$$\boldsymbol{x}_{\mathrm{E}} = \begin{pmatrix}\boldsymbol{p}_{\mathrm{E}}\\ \boldsymbol{\phi}_{\mathrm{E}}\end{pmatrix}\,, \quad \boldsymbol{x}_{\mathrm{I}} = \begin{pmatrix}\boldsymbol{p}_{\mathrm{I}}\\ \boldsymbol{\phi}_{\mathrm{I}}\end{pmatrix}\,, \tag{29.25}$$

where

$$\phi_E = \frac{1}{2}(\phi_{S,1} + \phi_{S,2}) , \quad \phi_I = \phi_{S,2} - \phi_{S,1} . \quad (29.26)$$

It must be remarked, however, that the definitions in (29.26) keep a clear geometric meaning only if the orientation displacements between the virtual stick frames are small. In this case, as shown in [29.11, 12], the corresponding operational space velocities, $\dot{x}_E$ and $\dot{x}_I$, correspond to v_E and v_I, respectively, with good approximation. Otherwise, if the orientation displacements become large, the variables defined in (29.26) do not represent any meaningful quantities, and other orientation representations have to be adopted, e.g., the unit quaternion (see Chap. 1 for a general introduction to quaternions and Sect. 29.3 for an application to cooperative robots kinematics).

Finally, by using (29.15) and (29.22), together with equations (29.3) and (29.6), the kinetostatic mappings between the force/velocities at the object and their counterparts in the joint space of the manipulators can be given

$$\tau = J_O^\top h_O , \quad (29.27)$$

$$v_O = J_O \dot{q} , \quad (29.28)$$

where $\tau = \left(\tau_1^\top \ \tau_2^\top\right)^\top$, $q = \left(q_1^\top \ q_2^\top\right)^\top$, and

$$J_O = U^\top J , \quad J = \begin{pmatrix} J_1 & O_6 \\ O_6 & J_2 \end{pmatrix} . \quad (29.29)$$

Formally similar mappings can be established in terms of operational space velocities $\dot{x}_E$ and $\dot{x}_I$ [29.11, 12] and the corresponding operational space forces/moments.

29.2.2 Multifingered Manipulation

Hereafter some connections between the field of multi-arm cooperative manipulation, described in this chapter, and multifingered manipulation, described in Chap. 28, are briefly outlined, focusing on the kinetostatics of the two classes of manipulation systems.

Both in the case of multi-arm cooperative systems and of multifingered manipulation, two or more manipulation structures grasp a commonly manipulated object.

Cooperative manipulation via multiple robotic arms is achieved by rigidly grasping the object (e.g., via rigid fixtures), and thus interaction takes place by exchanging both forces and moments at the grasp points. In other words, all the translational and rotational motion components are transmitted through the grasp points.

On the other side, when a multifingered hand manipulates an object, only some components of the motion are transmitted through the contact points. This is effectively modeled via properly defined constraint matrices, whose expression depends on the type of contact. In other words, constraint matrices act as filters selecting the components of the motion transmitted through each contact. In fact, as shown in Chap. 28, it is convenient to think of the object–finger contact point as twofold: a point on the tip of the finger and a point on the object. Hence, two generalized velocity vectors are defined for the i-th contact point (both expressed with respect to frame $\mathcal{T}_i$): the velocity of the contact point on the hand, $v^i_{h,i}$, and the velocity of the contact point on the object $v^i_{o,i}$. The corresponding dual generalized force vectors are $h^i_{h,i}$ and $h^i_{o,i}$, respectively. A contact model is then defined via the $(m_i \times 6)$ constraint matrix H_i, selecting m_i velocity components transmitted through the contact ($v^i_{t,i}$), i. e.,

$$v^i_{t,i} = H_i v^i_{h,i} = H_i v^i_{o,i} . \quad (29.30)$$

The relation dual to (29.30) is

$$H_i^\top h^i_{t,i} = h^i_{h,i} = h^i_{o,i} , \quad (29.31)$$

where $h^i_{t,i}$ is the vector of the transmitted generalized forces. Hence, (29.10) can be rewritten as

$$h_E = W_1 \overline{R}_1 H_1^\top h^1_{t,1} + W_2 \overline{R}_2 H_2^\top h^2_{t,2} , \quad (29.32)$$

where $\overline{R}_i = \mathrm{diag}\{R_i, R_i\}$. Hence, a conceptually similar kinetostatic analysis can be developed, leading to the concept of external and internal forces (and related kinematic quantities) as well.

However, it must be remarked that, while in the case of cooperative manipulators, tightly grasping the object, internal stresses are usually undesirable effects (unless controlled squeezing of a deformable object has to be achieved), in multifingered hands suitably controlled internal forces are useful to guarantee a firm grasp even in the presence of external loading applied to the object (see, e.g., the problem of form and force closure discussed in Chap. 28).

29.3 Cooperative Task Space

The task-oriented formulation for the definition of the *cooperative task space* originally proposed in [29.25,26] is here briefly reviewed. By starting from (29.23) and (29.24), this formulation defines directly the task variables in terms of the *absolute* and *relative* motion of the cooperative system, which can be directly computed from the position/orientation of the end-effector coordinate frames.

Let is define as *absolute coordinate frame*, $\mathcal{T}_a$, the frame whose position in the base frame is given by the position vector $\boldsymbol{p}_a$ (*absolute position*)

$$\boldsymbol{p}_\mathrm{a} = \frac{1}{2}(\boldsymbol{p}_1 + \boldsymbol{p}_2) . \tag{29.33}$$

The orientation of $\mathcal{T}_a$ with respect to the base frame (*absolute orientation*) is defined by means of the rotation matrix $\boldsymbol{R}_a$

$$\boldsymbol{R}_\mathrm{a} = \boldsymbol{R}_1 \boldsymbol{R}^1(\boldsymbol{k}^1_{21}, \vartheta_{21}/2) . \tag{29.34}$$

Of course, the sole absolute variables cannot uniquely specify the cooperative motion, since 12 variables are required, e.g., for a two-arm system. Hence, in order to devise a complete description of the system's configuration, the position/orientation of each manipulator relative to the other manipulators in the system must be considered. In detail, for a two-arm system, the *relative position* between the manipulators is defined as

$$\boldsymbol{p}_\mathrm{r} = \boldsymbol{p}_2 - \boldsymbol{p}_1 , \tag{29.35}$$

while the *relative orientation* between the two end-effector frames is expressed via the following rotation matrix

$$\boldsymbol{R}^1_\mathrm{r} = \boldsymbol{R}^\top_1 \boldsymbol{R}_2 = \boldsymbol{R}^1_2 . \tag{29.36}$$

The variables $\boldsymbol{p}_a$, $\boldsymbol{R}_a$, $\boldsymbol{p}_r$, and $\boldsymbol{R}^1_r$ define the *cooperative task space*. Remarkably, $\boldsymbol{R}_a$ and $\boldsymbol{R}^1_r$ coincide with $\boldsymbol{R}_E$ and $\boldsymbol{R}^1_I$, respectively.

It is worth pointing out a useful feature of the above defined cooperative task space formulation. In fact, it can be recognized that definitions (29.33–29.36) are not based on any special assumption on the nature of the manipulated object and/or the grasp. In other words, the cooperative task space variables can be effectively used to describe cooperative systems manipulating nonrigid objects and/or characterized by a nonrigid grasp. Also, the same task-space variables can be used to describe a pure motion coordination task, i.e., when the arms are required to perform a coordinated motion without physically interacting via a commonly manipulated object. When the manipulators hold a rigid object (or a deformable object for which deformations are not commanded), then relative position and orientation are to be kept constant. Otherwise, if a relative motion between the end-effectors is allowed (or commanded), then $\boldsymbol{p}^1_r$ and $\boldsymbol{R}^1_r$ may vary according to the effective (commanded) relative motion.

As shown in [29.25, 26], absolute

$$\dot{\boldsymbol{p}}_\mathrm{a} = \frac{1}{2}(\dot{\boldsymbol{p}}_1 + \dot{\boldsymbol{p}}_2) , \quad \boldsymbol{\omega}_\mathrm{a} = \frac{1}{2}(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2) , \tag{29.37}$$

and relative

$$\dot{\boldsymbol{p}}_\mathrm{r} = (\dot{\boldsymbol{p}}_2 - \dot{\boldsymbol{p}}_1) , \quad \boldsymbol{\omega}_\mathrm{r} = \boldsymbol{\omega}_2 - \boldsymbol{\omega}_1 , \tag{29.38}$$

linear and angular velocities can be readily derived, as well as their dual variables (i.e., absolute/relative forces and moments)

$$\boldsymbol{f}_\mathrm{a} = \boldsymbol{f}_1 + \boldsymbol{f}_2 , \quad \boldsymbol{n}_\mathrm{a} = \boldsymbol{n}_1 + \boldsymbol{n}_2 , \tag{29.39}$$

$$\boldsymbol{f}_\mathrm{r} = \frac{1}{2}(\boldsymbol{f}_2 - \boldsymbol{f}_1) , \quad \boldsymbol{n}_\mathrm{r} = \frac{1}{2}(\boldsymbol{n}_2 - \boldsymbol{n}_1) . \tag{29.40}$$

Kinetostatic mappings, analogous to those derived in the previous section, can be established between linear/angular velocities (force/moments) and their counterparts defined at the end-effectors (or joints) level of each manipulator [29.25, 26].

Remarkably, the variables defined by the symmetric formulation and those used in the task-oriented formulation are related via simple mappings. In fact, forces (angular velocities, orientation variables) always coincide in the two formulations, while moments (linear velocities, position variables) coincide only when the object reduces to a point, or when the kinematics of each manipulator is referred to the tip of the corresponding virtual stick.

As shown in the following example, in the case of planar cooperative systems, the definition of the cooperative task-space variables is straightforward.

Example 29.1 (cooperative task-space variables for a planar two-arm system): For a planar two-arm system the i-th end-effector coordinates can be defined via the (3×1) vector

$$\boldsymbol{x}_i = \begin{pmatrix} \boldsymbol{p}_i \\ \varphi_i \end{pmatrix} , \quad i = 1, 2 ,$$

where $\boldsymbol{p}_i$ is the (2×1) vector of the i-th end-effector position in the plane and φ_i is the angle describing its orientation (i. e., the rotation of the end-effector frame about an axis orthogonal to the plane). Hence, the task-space variables can be readily defined as

$$x_a = \frac{1}{2}(x_1 + x_2), \qquad (29.41)$$

$$x_r = x_2 - x_1, \qquad (29.42)$$

since the orientation of each end-effector is simply represented by an angle.

In the spatial case, the definition of the orientation variables in terms of Euler angles, as in (29.34) and (29.36), is somewhat critical; in fact, since $\mathcal{T}_1$ and $\mathcal{T}_2$ do not coincide in general, orientation displacements are likely to be large and definitions analogous to (29.26) are not correct. Hence, geometrically meaningful orientation representations have to be adopted, e.g., the unit quaternion (see Chap. 1). In detail, by following the approach in [29.26], the orientation variables can be defined as follows. Let

$$\mathcal{Q}_k^1 = \{\eta_k, \epsilon_k^1\} = \left\{\cos\frac{\vartheta_{21}}{4}, k_{21}^1 \sin\frac{\vartheta_{21}}{4}\right\} \qquad (29.43)$$

denote the unit quaternion extracted from $R^1(k_{21}^1, \vartheta_{21}/2)$; let, also, $\mathcal{Q}_1 = \{\eta_1, \epsilon_1\}$ and $\mathcal{Q}_2 = \{\eta_2, \epsilon_2\}$ denote the unit quaternions extracted from R_1 and R_2, respectively. Then, the absolute orientation can be expressed in terms of quaternion product as follows

$$\mathcal{Q}_a = \{\eta_a, \epsilon_a\} = \mathcal{Q}_1 * \mathcal{Q}_k^1, \qquad (29.44)$$

while the relative orientation can be expressed as the quaternion product

$$\mathcal{Q}_r^1 = \{\eta_r, \epsilon_r^1\} = \mathcal{Q}_1^{-1} * \mathcal{Q}_2, \qquad (29.45)$$

where $\mathcal{Q}_1^{-1} = \{\eta_1, -\epsilon_1\}$ (i. e., the conjugate of $\mathcal{Q}_1$) represents the unit quaternion extracted from $R_1^\top$.

29.4 Dynamics and Load Distribution

The equations of motion of the i-th manipulator in a cooperative manipulation system are given by

$$M_i(q_i)\ddot{q}_i + c_i(q_i, \dot{q}_i) = \tau_i - J_i^\top(q_i)h_i, \qquad (29.46)$$

where $M_i(q_i)$ is the symmetric positive-definite inertia matrix and $c_i(q_i, \dot{q}_i)$ is the vector of the forces/torques due to the centrifugal, Coriolis, gravity, and friction effects. The model can be expressed in compact form as

$$M(q)\ddot{q} + c(q, \dot{q}) = \tau - J^\top(q)h, \qquad (29.47)$$

where the matrices are block-diagonal (e.g., $M = \text{blockdiag}\{M_1, M_2\}$) and the vectors are stacked (e.g., $q = \left(q_1^\top\ q_2^\top\right)^\top$).

The object's motion is described by the classical Newton–Euler equations of rigid body

$$M_E(R_E)\dot{v}_E + c_E(R_E, \omega_E)v_E = h_E = Wh, \qquad (29.48)$$

where M_E is the inertia matrix of the object and c_E collects the nonlinear components of the inertial forces/moments (i. e., gravity, centrifugal and Coriolis forces/moments).

The above equations must be completed by imposing the closed-chain constraints arising from the kinematic coupling between the two manipulators through the commonly manipulated rigid object. The constraints can be expressed by imposing a null internal velocity vector in the mapping (29.21)

$$v_I = V_S^\top v_S = v_{S,1} - v_{S,2} = 0, \qquad (29.49)$$

which can be expressed, by using (29.8) and (29.22), in terms of end-effector velocities (where the notation $W_i^{-\top}$ stands for $(W_i^\top)^{-1}$)

$$V^\top v = W_1^{-\top} v_1 - W_2^{-\top} v_2 = 0, \qquad (29.50)$$

and, finally, in terms of joint velocities

$$V^\top J(q)\dot{q} = W_1^{-\top} J_1(q_1)\dot{q}_1 - W_2^{-\top} J_2(q_2)\dot{q}_2 = 0. \qquad (29.51)$$

Equations (29.47), (29.48), and (29.51) represent a constrained dynamical model of the cooperative system in the joint space; $N_1 + N_2$ generalized coordinates (i. e., q_1 and q_2) are related to each other by the six algebraic closed-chain constraints (29.51). This implies that the total number of degrees of freedom is $N_1 + N_2 - 6$ and the model has the form of a set of differential-algebraic equations.

29.4.1 Reduced-Order Models

The above derived dynamic model incorporates a set of closed-chain constraints, which reduces the number of independent generalized coordinates to $N_1 + N_2 - 6$.

Hence, it is expected that, by eliminating six equations via the constraints (29.51), a reduced-order model can be obtained. Early work on reduced-order modeling of closed chains can be, e.g., found in [29.58]. Later, in [29.59, 60], this problem has been tackled by first deriving, from (29.47), (29.48), and (29.51), a joint space model of the whole closed chain in the form

$$\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{D}_{\mathrm{C}}(\boldsymbol{q})\boldsymbol{\tau}\;, \qquad (29.52)$$

where $\boldsymbol{M}_{\mathrm{C}}$, $\boldsymbol{D}_{\mathrm{C}}$, and $\boldsymbol{c}_{\mathrm{C}}$ depend on the dynamics of the manipulators and the object, as well as on the geometry of the grasp. The above model can be integrated, once the joint torques vector $\boldsymbol{\tau}$ is specified over an assigned time interval, to solve for the joint variables $\boldsymbol{q}$ (*forward dynamics*). However, the model cannot be used to find $\boldsymbol{\tau}$ from assigned $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ (*inverse dynamics*), since the $((N_1+N_2)\times(N_1+N_2))$ matrix $\boldsymbol{D}_{\mathrm{C}}$ is not full rank [29.60], and thus the inverse dynamics problem turns out to be underspecified.

In order to find a reduced-order model, composed by N_1+N_2-6 equations, a $((N_1+N_2-6)\times 1)$ pseudovelocity vector has to be considered:

$$\boldsymbol{\nu}=\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\;, \qquad (29.53)$$

where the $((N_1+N_2-6)\times(N_1+N_2))$ matrix $\boldsymbol{B}(\boldsymbol{q})$ is selected so that $(\boldsymbol{A}^\top(\boldsymbol{q})\;\boldsymbol{B}^\top(\boldsymbol{q}))^\top$ is nonsingular and

$$\boldsymbol{A}(\boldsymbol{q})=\boldsymbol{W}_2^\top\,\boldsymbol{V}^\top(\boldsymbol{q})\;.$$

Then, the reduced-order model can be written in terms of the variables $\boldsymbol{q}$, $\boldsymbol{\nu}$, and $\dot{\boldsymbol{\nu}}$ as

$$\boldsymbol{\Sigma}^\top(\boldsymbol{q})\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\boldsymbol{\Sigma}(\boldsymbol{q})\,\dot{\boldsymbol{\nu}}+\boldsymbol{\Sigma}^\top(\boldsymbol{q})\boldsymbol{c}_{\mathrm{R}}(\boldsymbol{q},\boldsymbol{\nu})=\boldsymbol{\Sigma}^\top(\boldsymbol{q})\boldsymbol{\tau}\;, \qquad (29.54)$$

where $\boldsymbol{\Sigma}$ is an $((N_1+N_2)\times(N_1+N_2-6))$ matrix such that

$$\begin{pmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{pmatrix}^{-1}=(\boldsymbol{\Pi}\;\boldsymbol{\Sigma})\;,$$

and $\boldsymbol{c}_{\mathrm{R}}$ depends on $\boldsymbol{c}_{\mathrm{C}}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\Pi}$ [29.60]. The reduced-order model can be used for computing the forward dynamics; in this case, however, the problem of expressing a reduced set of *pseudocoordinates* related to $\boldsymbol{\nu}$ in the numerical integration has to be considered. Since $\boldsymbol{\Sigma}^\top$ is nonsquare, the inverse dynamics problem still admits infinitely many solutions in terms of $\boldsymbol{\tau}$. However, this does not prevent the application model (29.54) to cooperative manipulators control (see, e.g., the decoupled control architecture proposed in [29.59, 60]).

29.4.2 Load Distribution

The problem of load sharing in multi-arm robotic systems is that of distributing the load among the arms composing the system (e.g., a strong arm may share the load more than a weak one). This is possible because a multi-arm system has redundant actuators; in fact, if a robotic arm is equipped with the number of actuators strictly needed for supporting the load, no optimization of load distribution is possible. In this section, this problem is presented according to the results in [29.35–41].

We can introduce a load-sharing matrix in the framework adopted for presenting the kinematics of the cooperative manipulators. By replacing the Moore–Penrose inverse in equation (29.11) with a suitably defined generalized inverse, $\boldsymbol{W}_{\mathrm{S}}^{-}$, the following expression can be devised for the generalized forces at the tip of the virtual sticks

$$\boldsymbol{h}_{\mathrm{S}}=\boldsymbol{W}_{\mathrm{S}}^{-}\,\boldsymbol{h}_{\mathrm{E}}+\boldsymbol{V}_{\mathrm{S}}\,\boldsymbol{h}_{\mathrm{I}}'\;, \qquad (29.55)$$

where

$$\boldsymbol{W}_{\mathrm{S}}^{-}=\begin{pmatrix}\boldsymbol{L}\\ \boldsymbol{I}_6-\boldsymbol{L}\end{pmatrix}^\top\;, \qquad (29.56)$$

and the matrix $\boldsymbol{L}$ is the so-called load sharing matrix. It can be easily proved that the nondiagonal elements of $\boldsymbol{L}$ only yield a $\boldsymbol{h}_{\mathrm{S}}$ vector in the null space of $\boldsymbol{W}_{\mathrm{S}}$, that is, the space of internal forces/moments. Therefore, without losing generality, let us choose $\boldsymbol{L}$ such that

$$\boldsymbol{L}=\operatorname{diag}\{\boldsymbol{\lambda}\}\;, \qquad (29.57)$$

where the vector $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_6)^\top$ collects a set of constants λ_i representing the load-sharing coefficients.

The problem is that of properly tuning the load-sharing coefficients to ensure correct cooperative manipulation of the object by the arms. In order to answer this question, it must be noticed that, by combining equations (29.11) and (29.55), the following relation can be obtained

$$\boldsymbol{h}_{\mathrm{I}}=\boldsymbol{V}_{\mathrm{S}}^{\dagger}\big(\boldsymbol{W}_{\mathrm{S}}^{-}-\boldsymbol{W}_{\mathrm{S}}^{\dagger}\big)\boldsymbol{h}_{\mathrm{E}}+\boldsymbol{h}_{\mathrm{I}}'\;, \qquad (29.58)$$

which, bearing in mind that only $\boldsymbol{h}_{\mathrm{E}}$ and $\boldsymbol{h}_{\mathrm{S}}$ are really existing forces/moments, indicates that

- $\boldsymbol{h}_{\mathrm{I}}$, $\boldsymbol{h}_{\mathrm{I}}'$, and λ_i can be thought of as *artificial* parameters, introduced for better understanding of the manipulation process;
- $\boldsymbol{h}_{\mathrm{I}}'$ and λ_i are not independent; the concept of internal forces/moments and the concept of load sharing are mathematically mixed with each other.

Therefore, tuning the load-sharing coefficients or choosing suitable internal forces/moments is strictly equivalent from the mathematical (and, also, from the performance) point of view. Only one variable among $\boldsymbol{h}_\mathrm{I}$, $\boldsymbol{h}'_\mathrm{I}$, and $\boldsymbol{\lambda}$ is independent; hence, any of those redundant parameters, to be optimized for load sharing, can be exploited. This is more generally stated in [29.38, 39]. In [29.41], internal forces/moments $\boldsymbol{h}_\mathrm{I}$ are tuned, both for the sake of simplicity and for consistency with the adopted control laws.

A very relevant problem related to load sharing is that of robust holding, i. e., the problem of determining forces/moments applied to object by the arms, $\boldsymbol{h}_\mathrm{S}$, required to keep the grasp even in the presence of disturbing external forces/moments. This problem can be solved by tuning the internal forces/moments (or the load-sharing coefficients, of course) and is addressed, e.g., in [29.40], where conditions to keep the grasp are expressed via end-effector forces/moments. Namely, by replacing $\boldsymbol{h}_\mathrm{S}$ in (29.55) into the equations expressing the grasp conditions, a set of linear inequalities for both $\boldsymbol{h}'_\mathrm{I}$ and $\boldsymbol{\lambda}$ are obtained:

$$\boldsymbol{A}_L \boldsymbol{h}'_\mathrm{I} + \boldsymbol{B}_L \boldsymbol{\lambda} < \boldsymbol{c}_L \, , \tag{29.59}$$

where $\boldsymbol{A}_L$ and $\boldsymbol{B}_L$ are 6×6 matrices and $\boldsymbol{c}_L$ is a (6×1) vector. In [29.40], a solution $\boldsymbol{\lambda}$ for the inequality is obtained heuristically. The above inequality can be transformed into another inequality expressed with respect to $\boldsymbol{h}_\mathrm{I}$, of course; however, $\boldsymbol{\lambda}$ is fitter to such heuristic algorithm because it can be understood intuitively. The same problem may be solved mathematically, by introducing an objective function to be optimized; in this way, the problem is recast in the framework of mathematical programming. To this purpose, let us choose a quadratic cost function of $\boldsymbol{h}_\mathrm{I}$ to be minimized

$$f = \boldsymbol{h}_\mathrm{I}^\top \boldsymbol{Q} \boldsymbol{h}_\mathrm{I} \, , \tag{29.60}$$

where $\boldsymbol{Q}$ is a (6×6) positive-definite matrix. The above cost function can be seen as a (pseudo)energy to be dissipated by the joint actuators: i. e., the arms dissipate electrical energy in the actuators to yield the internal forces/moments $\boldsymbol{h}_\mathrm{I}$. A solution to the quadratic programming problem (29.60) can be found, e.g., in [29.41].

For further insights on the problem of robust holding in the framework of multifingered manipulation, the reader is referred to Chap. 28.

Interestingly, in [29.36] the problem of load distribution is formulated in such a way to take into account manipulators dynamics at the joint level. This approach allows the expression of the load-sharing problem directly in terms of joints actuators torques; at this level, different subtask performance indexes can be used, in a similar way as was done to solve the inverse kinematics of redundant manipulators, to achieve load distribution solutions.

29.5 Task-Space Analysis

As for single-arm robotic systems, a major issue in cooperative manipulation is that of task-space performance evaluation via the definition of suitable manipulability ellipsoids. These concepts have been extended to multiarm robotic systems in [29.30]. Namely, by exploiting the kinetostatic formulation in Sect. 29.3, velocity and force manipulability ellipsoids are defined by regarding the whole cooperative system as a mechanical transformer from the joint space to the cooperative task space. Since the construction of force/velocity ellipsoids involves nonhomogeneous quantities (i. e., forces and moments, linear and angular velocities), special attention must be paid to the definition of such concepts [29.14, 61, 62]. Also, as for the ellipsoids involving internal forces, a major issue is that of physically meaningful parametrization of the internal forces (see, e.g., the work in [29.15, 16]).

In detail, by following the approach in [29.30], the *external force manipulability ellipsoid* can be defined by the following scalar equation

$$\boldsymbol{h}_\mathrm{E}^\top (\boldsymbol{J}_\mathrm{E} \boldsymbol{J}_\mathrm{E}^\top) \boldsymbol{h}_\mathrm{E} = 1 \, , \tag{29.61}$$

where $\boldsymbol{J}_\mathrm{E} = \boldsymbol{W}^{\dagger\top} \boldsymbol{J}$. The *external velocity manipulability ellipsoid* is defined dually by the following scalar equation

$$\boldsymbol{v}_\mathrm{E}^\top (\boldsymbol{J}_\mathrm{E} \boldsymbol{J}_\mathrm{E}^\top)^{-1} \boldsymbol{v}_\mathrm{E} = 1 \, . \tag{29.62}$$

In the case of dual-arm systems the *internal force manipulability ellipsoid* can be defined as

$$\boldsymbol{h}_\mathrm{I}^\top (\boldsymbol{J}_\mathrm{I} \boldsymbol{J}_\mathrm{I}^\top) \boldsymbol{h}_\mathrm{I} = 1 \, , \tag{29.63}$$

where $\boldsymbol{J}_\mathrm{I} = \boldsymbol{V}^\top \boldsymbol{J}$. Also, the *internal velocity ellipsoid* can be defined, via kinetostatic duality, as

$$\boldsymbol{v}_\mathrm{I}^\top (\boldsymbol{J}_\mathrm{I} \boldsymbol{J}_\mathrm{I}^\top)^{-1} \boldsymbol{v}_\mathrm{I} = 1 \, . \tag{29.64}$$

The internal force/velocity ellipsoids can be defined in the case of cooperative systems composed of more than two manipulators by considering one pair of interacting end-effectors at a time [29.30].

The manipulability ellipsoids can be seen as performance measures aimed at determining the arms' attitude to cooperate in a given system's configuration. Also, as for the single-arm systems, the manipulability ellipsoids can be used to determine optimal postures for redundant multi-arm systems.

Besides the above described approach, two other main approaches have been proposed to analyze the manipulability of cooperative multi-arm systems: the task-oriented manipulability measure [29.31] and polytopes [29.32]. Moreover, in [29.33] a systematic approach to perform dynamic analysis of multi-arm systems is presented. Namely, the concept of dynamic manipulability ellipsoid is extended to multi-arm systems, and indexes of the system's capability of generating object accelerations along assigned task-space directions are devised.

29.6 Control

When a cooperative multi-arm system is employed for the manipulation of a common object, it is important to control both the absolute motion of the held object and the internal stresses applied to it. Hence, most of the control approaches to cooperative robotic systems can be classified as force/motion control schemes, in that they decompose the control action in a motion control loop, aimed at tracking of the desired object motion, and a force control loop, aimed at controlling the internal loading of the object.

Early approaches to the control of cooperative systems are based on the *master/slave* concept [29.2]. Namely, the cooperative system is decomposed into:

- a master arm, which is in charge of imposing the absolute motion of the object; hence, the master arm is position controlled so as to achieve accurate and robust tracking of position/orientation reference trajectories, in the face of external disturbances (e.g., forces due to the interaction with the other cooperating arms); in other words, the master arm is controlled so as to have a *stiff* behavior;
- the slave arms, which are force controlled so as to achieve a *compliant* behavior with respect to the interaction forces; hence, it is expected that the slave arms are capable of following (as smoothly as possible) the motion imposed by the master arm.

A natural evolution of the above approach is the so-called *leader–follower* [29.58], where the follower arms reference motion is computed via closed-chain constraints.

However, such approaches suffered from implementation issues, mainly due to the fact that the compliance of the slave arms has to be very large, so as to follow the motion imposed by the master arm smoothly. Also, a difficulty arises when the roles of master and slave have to be assigned to the arms for a given cooperative task, since the master/slave modes may need to be dynamically changed during the task execution.

Hence, a more natural non-master/slave approach has been pursued later, where the cooperative system is seen as a whole. Namely, the reference motion of the object is used to determine the motion of all the arms in the system and the interaction forces acting at each end-effector are fed back so as to be directly controlled. To this aim, the mappings between the forces and velocities at the end-effector of each manipulator and their counterparts at the manipulated object are to be considered in the design of the control laws.

29.6.1 Hybrid Control

In [29.11, 12] a non-master/slave approach has been proposed, based on the well-known scheme proposed by Raibert and Craig for the robot/environment interaction control of single-arm systems (see Chap. 7). The operational space vector for the hybrid position/force control is defined via the following variables

$$\boldsymbol{x}_{\mathrm{O}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{E}} \\ \boldsymbol{x}_{\mathrm{I}} \end{pmatrix} , \qquad (29.65)$$

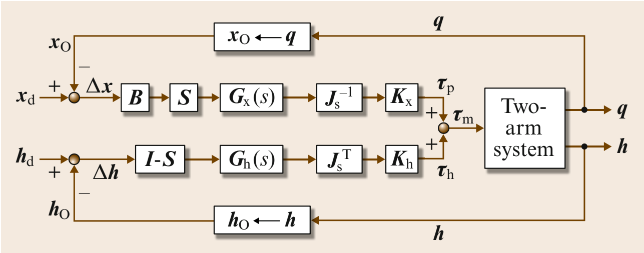

Fig. 29.2 A hybrid position/force control scheme

where $\boldsymbol{x}_\mathrm{E}$, $\boldsymbol{x}_\mathrm{I}$ are the operational space vectors, defined by specifying the orientation via a minimal set of orientation angles (e.g., Euler angles). The generalized forces vector to be considered is

$$\boldsymbol{h}_\mathrm{O} = \begin{pmatrix} \boldsymbol{h}_\mathrm{E} \\ \boldsymbol{h}_\mathrm{I} \end{pmatrix} . \tag{29.66}$$

The organization of the control scheme is shown diagrammatically in Fig. 29.2. The suffixes d and m represent the desired value and the control command, respectively. The command vector $\boldsymbol{\tau}_\mathrm{m}$ to the actuators of the two arms is given by two contributions:

$$\boldsymbol{\tau}_\mathrm{m} = \boldsymbol{\tau}_\mathrm{p} + \boldsymbol{\tau}_\mathrm{h} . \tag{29.67}$$

The first term, $\boldsymbol{\tau}_\mathrm{p}$, is the command vector for the position control and is given by

$$\boldsymbol{\tau}_\mathrm{p} = \boldsymbol{K}_\mathrm{x} \boldsymbol{J}_\mathrm{s}^{-1} \boldsymbol{G}_\mathrm{x}(s) \boldsymbol{S} \boldsymbol{B} (\boldsymbol{x}_{\mathrm{O,d}} - \boldsymbol{x}_\mathrm{O}) , \tag{29.68}$$

while $\boldsymbol{\tau}_\mathrm{h}$ is the command vector for the force control

$$\boldsymbol{\tau}_\mathrm{h} = \boldsymbol{K}_\mathrm{h} \boldsymbol{J}_\mathrm{s}^\top \boldsymbol{G}_\mathrm{h}(s)(\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{h}_{\mathrm{O,d}} - \boldsymbol{h}_\mathrm{O}) . \tag{29.69}$$

The matrix $\boldsymbol{B}$ transforms the errors on the orientation angles into equivalent rotation vectors. The matrix $\boldsymbol{J}_\mathrm{s}$ is the Jacobian matrix that transforms the joint velocity $\dot{\boldsymbol{q}}$ into the task-space velocity $\boldsymbol{v}_\mathrm{O}$. The matrix operators $\boldsymbol{G}_\mathrm{x}(s)$ and $\boldsymbol{G}_\mathrm{h}(s)$ represent position and force control laws, respectively. The gain matrices $\boldsymbol{K}_\mathrm{x}$ and $\boldsymbol{K}_\mathrm{h}$ are assumed to be diagonal; their diagonal elements convert velocity and force commands into actuator commands, respectively. The matrix $\boldsymbol{S}$ selects the position-controlled variables; it is diagonal and its diagonal entries take the values of 1 or 0; namely, the i-th workspace coordinate is position controlled if the i-th diagonal element of $\boldsymbol{S}$ is 1, while it is force controlled if it is 0. Finally, $\boldsymbol{I}$ is the identity matrix having the same dimensions as $\boldsymbol{S}$, while $\boldsymbol{q}$ and $\boldsymbol{h}$ are the vectors of measured joint variables and measured end-effector generalized forces, respectively.

29.6.2 PD Force/Motion Control

In [29.17] a Lyapunov-based approach is pursued to devise force/position PD-type control laws. Namely, the joints torques inputs to each arm are computed as the combination of two contributions:

$$\boldsymbol{\tau}_\mathrm{m} = \boldsymbol{\tau}_\mathrm{p} + \boldsymbol{\tau}_\mathrm{h} , \tag{29.70}$$

where $\boldsymbol{\tau}_\mathrm{p}$ is a PD-type term (eventually including a feedback/feedforward model-based compensation term), taking care of position control, while $\boldsymbol{\tau}_\mathrm{h}$ is in charge of internal forces/moments control.

Namely, the PD and model-based terms can be computed at the joint level

$$\boldsymbol{\tau}_\mathrm{p} = \boldsymbol{K}_\mathrm{p} \boldsymbol{e}_\mathrm{q} - \boldsymbol{K}_\mathrm{d} \dot{\boldsymbol{q}} + \boldsymbol{g} + \boldsymbol{J}^\top \boldsymbol{W}^\dagger \boldsymbol{g}_\mathrm{E} , \tag{29.71}$$

where $\boldsymbol{e}_\mathrm{q} = \boldsymbol{q}_\mathrm{d} - \boldsymbol{q}$, $\boldsymbol{q}_\mathrm{d}$ is the vector of desired joint variables, $\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{d}$ are positive-definite matrix gains, $\boldsymbol{g}$ is the vector of the gravitational forces/torques acting at the manipulators joints, $\boldsymbol{g}_\mathrm{E}$ is the vector of gravity forces/moments at the manipulated object.

Since the cooperative task is usually assigned in terms of absolute and relative motion, the equivalent end-effector desired trajectories are to be computed by using the closed-chain constraints, as in the following example.

Example 29.2 (computation of desired trajectories for a planar two-arm system): For the planar two-arm system, the desired trajectories defining the cooperative task are assigned by specifying the absolute, $\boldsymbol{x}_{\mathrm{a,d}}(t)$, and relative, $\boldsymbol{x}_{\mathrm{r,d}}(t)$, desired motion. Then, the corresponding end-effectors desired trajectories can be computed as

$$\boldsymbol{x}_{1,\mathrm{d}}(t) = \boldsymbol{x}_{\mathrm{a,d}}(t) - \frac{1}{2} \boldsymbol{x}_{\mathrm{r,d}}(t) , \tag{29.72}$$

$$\boldsymbol{x}_{2,\mathrm{d}}(t) = \boldsymbol{x}_{\mathrm{a,d}}(t) + \frac{1}{2} \boldsymbol{x}_{\mathrm{r,d}}(t) , \tag{29.73}$$

where (29.41) and (29.42) have been exploited.

In the spatial case (29.72, 29.73) do not hold, since the absolute and relative orientation must be expressed in terms of geometrically meaningful quantities, e.g., via (29.44, 29.45). However, the same approach may be pursued at the expense of a slightly more complex expressions of the above formulas [29.26].

Once, the desired position/orientation for each end-effector has been obtained, the inverse kinematics of each arm has to be computed to provide the desired joint trajectories, $\boldsymbol{q}_\mathrm{d}$, to the control loop; to this aim, e.g., numerical inverse kinematics algorithms may be employed (see Chaps. 1 and 11).

Also, a PD-type control law can be expressed in terms of end-effector variables, i. e.,

$$\boldsymbol{\tau}_\mathrm{p} = \boldsymbol{J}^\top (\boldsymbol{K}_\mathrm{p} \boldsymbol{e} - \boldsymbol{K}_\mathrm{v} \boldsymbol{v}) - \boldsymbol{K}_\mathrm{d} \dot{\boldsymbol{q}} + \boldsymbol{g} + \boldsymbol{J}^\top \boldsymbol{W}^\dagger \boldsymbol{g}_\mathrm{E} , \tag{29.74}$$

where $\boldsymbol{e}$ is the tracking error computed in terms of end-effector position/orientation variables, $\boldsymbol{v}$ is the vector collecting the end-effector velocities, and $\boldsymbol{K}_\mathrm{p}$, $\boldsymbol{K}_\mathrm{v}$, and $\boldsymbol{K}_\mathrm{d}$ are positive-definite matrix gains.

Finally, the same concept may be exploited to design a PD control law directly in the object space

$$\boldsymbol{\tau}_\mathrm{p} = \boldsymbol{J}^\top \boldsymbol{W}^\dagger (\boldsymbol{K}_\mathrm{p}\boldsymbol{e}_\mathrm{E} - \boldsymbol{K}_\mathrm{v}\boldsymbol{v}_\mathrm{E}) - \boldsymbol{K}_\mathrm{d}\dot{\boldsymbol{q}} + \boldsymbol{g} + \boldsymbol{J}^\top \boldsymbol{W}^\dagger \boldsymbol{g}_\mathrm{E} , \tag{29.75}$$

where $\boldsymbol{e}_\mathrm{E}$ is the tracking error computed in terms of object absolute position/orientation variables, $\boldsymbol{v}_\mathrm{E}$ is the object's generalized velocity vector, and $\boldsymbol{K}_\mathrm{p}$, $\boldsymbol{K}_\mathrm{v}$, and $\boldsymbol{K}_\mathrm{d}$ are positive-definite matrix gains.

The internal force control term can instead be designed as follows

$$\boldsymbol{\tau}_\mathrm{h} = \boldsymbol{J}^\top \boldsymbol{V}\boldsymbol{h}_{\mathrm{I,c}} , \tag{29.76}$$

where

$$\boldsymbol{h}_{\mathrm{I,c}} = \boldsymbol{h}_{\mathrm{I,d}} + \boldsymbol{G}_\mathrm{h}(s)(\boldsymbol{h}_{\mathrm{I,d}} - \boldsymbol{h}_\mathrm{I}) , \tag{29.77}$$

$\boldsymbol{G}_\mathrm{h}(s)$ is a matrix operator representing a strictly proper linear filter, such that $\boldsymbol{I} - \boldsymbol{G}_\mathrm{h}(s)$ has zeros only in the left half plane, and $\boldsymbol{h}_{\mathrm{I,d}}$ is the vector of desired internal forces; the internal force vector can be computed from the vector of measured end-effector forces as $\boldsymbol{h}_\mathrm{I} = \boldsymbol{V}^\dagger \boldsymbol{h}$. A particularly simple choice, ensuring a null error at steady state, is given by

$$\boldsymbol{G}_\mathrm{h}(s) = \frac{1}{s}\boldsymbol{K}_\mathrm{h} ,$$

where $\boldsymbol{K}_\mathrm{h}$ is a positive-definite matrix. Remarkably, when an infinitely rigid object/grasp is considered, preprocessing of the force error via a strictly proper filter is needed to ensure closed-loop stability [29.17]; e.g., if a simple proportional feedback is adopted (i. e., $\boldsymbol{G}_\mathrm{h}(s) = \boldsymbol{K}_\mathrm{h}$), the closed loop will be unstable in the presence of an arbitrary small time delay, unless a small (namely, smaller than 1) force gain is adopted. In practice, the closed kinematic chain will be characterized by some elasticity (e.g., due to grippers, end-effector force/torques sensors, joints); in this case, the product between the control gain $\boldsymbol{K}_\mathrm{h}$ and the stiffness of the flexible components has to be chosen sufficiently small to ensure stability.

An interesting extension of the above approach has been given in [29.26, 28], where kinetostatic filtering of the control action is performed so as to filter all the components of the control input which contribute to internal stresses at the object. Namely, the control law (29.71) can be modified by weighting the proportional term ($\boldsymbol{K}_\mathrm{p}\,\boldsymbol{e}_\mathrm{q}$) via the filtering matrix

$$\boldsymbol{\phi} = \boldsymbol{J}^\top (\boldsymbol{W}^\dagger \boldsymbol{W} + \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\dagger)\boldsymbol{J}^{-\top} ,$$

where the (6×6) diagonal matrix $\boldsymbol{\Sigma} = \mathrm{diag}\{\sigma_i\}$ weights the components of $\boldsymbol{J}^{-\top}\boldsymbol{K}_\mathrm{p}\boldsymbol{e}_\mathrm{q}$ in each direction of the subspace of the internal forces via the constant values $0 \le \sigma_i \le 1$. In detail, if $\boldsymbol{\Sigma} = \boldsymbol{O}_6$ then all these components are completely canceled from the control action, while the choice $\boldsymbol{\Sigma} = \boldsymbol{I}_6$ leads to the control law (29.71) without kinetostatic filtering. In a similar way, the control laws (29.74) and (29.75) can be modified so as to introduce proper kinetostatic filtering of $\boldsymbol{K}_\mathrm{p}\,\boldsymbol{e}$ and $\boldsymbol{K}_\mathrm{p}\,\boldsymbol{e}_\mathrm{E}$, respectively.

29.6.3 Feedback Linearization Approaches

A further improvement of the PD-plus-gravity compensation control approach has been achieved by introducing a full model compensation, so as to achieve feedback/feedforward linearization of the closed-loop system. The feedback linearization approach formulated at the operational space level is the base for the so-called *augmented object* approach [29.63, 64]. In this approach the system is modeled in the operational space as a whole, by suitably expressing its inertial properties via a single augmented inertia matrix $\boldsymbol{M}_\mathrm{O}$. Hence, the dynamics of the cooperative system in the operational space can be written as

$$\boldsymbol{M}_\mathrm{O}(\boldsymbol{x}_\mathrm{E})\ddot{\boldsymbol{x}}_\mathrm{E} + \boldsymbol{c}_\mathrm{O}(\boldsymbol{x}_\mathrm{E}, \dot{\boldsymbol{x}}_\mathrm{E}) = \boldsymbol{h}_\mathrm{E} . \tag{29.78}$$

In (29.78), $\boldsymbol{M}_\mathrm{O}$ and $\boldsymbol{c}_\mathrm{O}$ are the operational space terms modeling, respectively, the inertial properties of the whole system (manipulators and object) and the Coriolis, centrifugal, friction, and gravity terms.

In the framework of feedback linearization (formulated in the operational), the problem of controlling the internal forces can be solved, e.g., by resorting to the *virtual linkage* model [29.16] or according to the scheme proposed in [29.29], i. e.,

$$\begin{aligned}\boldsymbol{\tau} = {} & \boldsymbol{J}^\top \boldsymbol{W}^\dagger \left[\boldsymbol{M}_\mathrm{O}\big(\ddot{\boldsymbol{x}}_{\mathrm{E,d}} + \boldsymbol{K}_\mathrm{v}\dot{\boldsymbol{e}}_\mathrm{E} + \boldsymbol{K}_\mathrm{p}\boldsymbol{e}_\mathrm{E}\big) + \boldsymbol{c}_\mathrm{O}\right] \\ & + \boldsymbol{J}^\top \boldsymbol{V}\Big(\boldsymbol{h}_{\mathrm{I,d}} + \boldsymbol{K}_\mathrm{h}\int(\boldsymbol{h}_{\mathrm{I,d}} - \boldsymbol{h}_\mathrm{I})\Big) .\end{aligned} \tag{29.79}$$

The above control law yields a linear and decoupled closed-loop dynamics

$$\begin{aligned}&\ddot{\boldsymbol{e}}_\mathrm{E} + \boldsymbol{K}_\mathrm{v}\dot{\boldsymbol{e}}_\mathrm{E} + \boldsymbol{K}_\mathrm{p}\boldsymbol{e}_\mathrm{E} = \boldsymbol{0} \\ &\tilde{\boldsymbol{h}}_\mathrm{I} + \boldsymbol{K}_\mathrm{h}\int \tilde{\boldsymbol{h}}_\mathrm{I}\,\mathrm{d}t = \boldsymbol{0} ,\end{aligned} \tag{29.80}$$

where $\tilde{\boldsymbol{h}}_\mathrm{I} = \boldsymbol{h}_{\mathrm{I,d}} - \boldsymbol{h}_\mathrm{I}$. Hence, the closed-loop dynamics guarantees asymptotically vanishing motion and force errors.

29.6.4 Impedance Control

An alternative control approach can be pursued based on the well-known impedance concept (see Chap. 7). In fact, when a manipulation system interacts with an external environment and/or other manipulators, large values of the contact forces and moments can be avoided by enforcing a compliant behavior, with suitable dynamic features, of the robotic system. Impedance control schemes have been proposed in the case of cooperative manipulation to control object/environment interaction forces [29.21] or internal forces [29.22]. More recently, an impedance scheme for the control of both external forces and internal forces has been proposed [29.65].

In detail, the impedance scheme in [29.21] enforces the following mechanical impedance behavior between the object displacements and the forces acting on object due to interaction with the environment:

$$\boldsymbol{M}_{\mathrm{E}}\tilde{\boldsymbol{a}}_{\mathrm{E}} + \boldsymbol{D}_{\mathrm{E}}\tilde{\boldsymbol{v}}_{\mathrm{E}} + \boldsymbol{K}_{\mathrm{e}}\boldsymbol{e}_{\mathrm{E}} = \boldsymbol{h}_{\mathrm{env}} , \tag{29.81}$$

where:

- $\boldsymbol{e}_{\mathrm{E}}$ represents the vector of the object's displacements between the desired and actual pose,
- $\tilde{\boldsymbol{v}}_{\mathrm{E}}$ is the difference between the object's desired and actual generalized velocities,
- $\tilde{\boldsymbol{a}}_{\mathrm{E}}$ is the difference between the object's desired and actual generalized accelerations,

and $\boldsymbol{h}_{\mathrm{env}}$ is the generalized force acting on object due to the interaction with the environment. The impedance dynamics is characterized in terms of given positive-definite mass ($\boldsymbol{M}_{\mathrm{E}}$), damping ($\boldsymbol{D}_{\mathrm{E}}$), and stiffness ($\boldsymbol{K}_{\mathrm{E}}$) matrices to be properly chosen so as to achieve the desired compliant behavior of the object.

As for the pose displacements used in (29.81), special attention has to be paid to the orientation variables.

Example 29.3 (external impedance for a planar two-arm system): For the planar two-arm system the quantities in (29.81) can be defined in a straightforward way. Namely,

$$\boldsymbol{e}_{\mathrm{E}} = \boldsymbol{x}_{\mathrm{E,d}} - \boldsymbol{x}_{\mathrm{E}} , \tag{29.82}$$

$$\tilde{\boldsymbol{v}}_{\mathrm{E}} = \dot{\boldsymbol{x}}_{\mathrm{E,d}} - \dot{\boldsymbol{x}}_{\mathrm{E}} , \tag{29.83}$$

$$\tilde{\boldsymbol{a}}_{\mathrm{E}} = \ddot{\boldsymbol{x}}_{\mathrm{E,d}} - \ddot{\boldsymbol{x}}_{\mathrm{E}} , \tag{29.84}$$

where $\boldsymbol{x}_{\mathrm{E}}$ is the (3×1) vector collecting the object's position and the orientation, while $\boldsymbol{x}_{\mathrm{E,d}}$ represents its desired counterpart. Hence, $\boldsymbol{M}_{\mathrm{E}}$, $\boldsymbol{D}_{\mathrm{E}}$, and $\boldsymbol{K}_{\mathrm{E}}$ are (3×3) matrices.

In the spatial case, orientation displacements cannot be defined as in (29.82), and geometrically meaningful orientation representations (i. e., rotation matrices and/or angle/axis representations) or in terms of operational space variables (i. e., differences between operational space vectors) have to be adopted [29.65].

The impedance scheme in [29.22], enforces a mechanical impedance behavior between the i-th end-effector displacements and the internal forces, i. e.,

$$\boldsymbol{M}_{\mathrm{I},i}\,\tilde{\boldsymbol{a}}_i + \boldsymbol{D}_{\mathrm{I},i}\,\tilde{\boldsymbol{v}}_i + \boldsymbol{K}_{\mathrm{I},i}\,\boldsymbol{e}_i = \boldsymbol{h}_{\mathrm{I},i} , \tag{29.85}$$

where

- $\boldsymbol{e}_i$ is the vector expressing the displacement of the i-th end-effector between the desired and actual pose,
- $\tilde{\boldsymbol{v}}_i$ is the vector expressing the difference between the desired and actual velocities of the i-th end-effector,
- $\tilde{\boldsymbol{a}}_i$ is the vector expressing the difference between the desired and actual accelerations of the i-th end-effector,

and $\boldsymbol{h}_{\mathrm{I},i}$ is the contribution of the i-th end-effector to the internal force, i. e., the i-th component of the vector $\boldsymbol{V}\boldsymbol{V}^{\dagger}\boldsymbol{h}$. Again, the impedance dynamics is characterized in terms of given positive-definite mass ($\boldsymbol{M}_{\mathrm{I},i}$), damping ($\boldsymbol{D}_{\mathrm{I},i}$), and stiffness ($\boldsymbol{K}_{\mathrm{I},i}$) matrices, to be properly chosen so as to achieve a suitable compliant behavior of the end-effectors with respect to internal forces.

Example 29.4 (Internal impedance for a planar two-arm system): For the planar two-arm system the quantities in (29.85) can be defined in a straightforward way as well

$$\boldsymbol{e}_i = \boldsymbol{x}_{i,d} - \boldsymbol{x}_i , \tag{29.86}$$

$$\tilde{\boldsymbol{v}}_i = \dot{\boldsymbol{x}}_{i,d} - \dot{\boldsymbol{x}}_i , \tag{29.87}$$

$$\tilde{\boldsymbol{a}}_i = \ddot{\boldsymbol{x}}_{i,d} - \ddot{\boldsymbol{x}}_i , \tag{29.88}$$

where $\boldsymbol{x}_i$ is the (3×1) vector collecting the position in the plane and the orientation angle of the i-th end-effector, while $\boldsymbol{x}_{i,d}$ represents its desired counterpart. Again, $\boldsymbol{M}_{\mathrm{I},i}$, $\boldsymbol{D}_{\mathrm{I},i}$, and $\boldsymbol{K}_{\mathrm{I},i}$ are (3×3) matrices.

As for the spatial case, orientation displacements in (29.85) cannot be defined as in (29.86), and geometrically meaningful orientation representations (i. e., rotation matrices and/or angle/axis representations) or in terms of operational space variables (i. e., differences between operational space vectors) have to be adopted [29.65].

The above two approaches have been combined in [29.65], where two control loops are designed to enforce an impedance behavior both at the object level (external forces) and at the end-effector level (internal forces).

29.7 Conclusions and Further Reading

In this chapter, the fundamentals of cooperative manipulation have been presented. A historical overview of the research on cooperative manipulation has been provided. The kinematics and dynamics of robotic arms cooperatively manipulating a tightly grasped rigid object are considered. Special topics, such as the definition of a cooperative task space and the problem of load distribution, have also been touched on. Then, the main control approaches for cooperative systems have been discussed. A few advanced topics related to control of cooperative robots, i. e., advanced nonlinear control as well as the modeling and control of cooperative systems including flexible elements, will be briefly outlined in the following.

Some of the first attempts to cope with uncertainties and disturbances in cooperative robots control focused on adaptive strategies [29.23,24], in which the unknown parameters are estimated online, based on a suitable linear-in-the-parameters model of the uncertainties. In detail, the approach in [29.23] is aimed at controlling the object motion, the interaction force due to the contact object/environment, and the internal forces; the adaptive control law estimates the unknown model parameters, both of the manipulators and of the object, on the basis of suitable error equations. In [29.24], the adaptive control concept is used to design a decentralized scheme, i. e., a centralized coordinator is not used, for redundant cooperative manipulators.

A recently developed control framework for cooperative systems is the so-called synchronization control [29.49, 50]; in this approach the control problem is formulated in terms of suitably defined errors accounting for motion synchronization between the manipulators involved in the cooperative task. Namely, the key idea in [29.49] is to ensure tracking of the desired trajectory assigned to each manipulator, while synchronizing its motion with other manipulators motion. In [29.50], the problem of synchronizing the motion of multiple manipulators systems is solved by using only position measurements; the synchronization controller consists of a feedback control law and a set of nonlinear observers, and synchronization is ensured by suitably defining the coupling errors between the manipulators' motions.

Recently, research efforts have been spent on intelligent control [29.51, 52]. Namely, in [29.51], a semi-decentralized adaptive fuzzy control scheme, with $\mathcal{H}_\infty$ performance in motion and internal force tracking, is proposed; the controller of each robot consists of two parts: a model-based adaptive controller and an adaptive fuzzy logic controller; the model-based adaptive controller handles the nominal dynamics, including a purely parametric uncertainties model, while the fuzzy logic controller is aimed at counteracting the effect of unstructured uncertainties and external disturbances. In [29.52], a decentralized adaptive fuzzy control scheme is proposed, where the control law makes use of a multi-input multi-output fuzzy logic engine and a systematic online adaptation mechanism.

Finally it is worth mentioning the efforts invested in the investigation of control strategies using partial state feedback, i. e., only joints position and end-effector forces are fed back to the controller. A recent contribution was provided by the work in [29.53], in which a decentralized control algorithm that achieves asymptotic tracking of desired positions and forces by using a nonlinear observer for velocities was proposed.

Flexibility in a cooperative system may arise, e.g., due to the use of nonrigid grippers to grasp the manipulated object. In fact, the adoption of compliant grippers allows large internal forces to be avoided and, at the same time, achieves safe manipulation of the object, even in the presence of failures and unpredicted contacts with the external environment. In detail, the work in [29.57] develops a non-model-based decentralized control scheme. Namely, a proportional–derivative (PD) position feedback control scheme with gravity compensation, is designed; the PD scheme is capable of regulating the position/orientation of the manipulated object and, simultaneously, achieves damping of the vibrations induced by the compliant grippers; also, a hybrid scheme is adopted to control internal forces along the directions in which the compliance of grippers is too low to ensure limited internal stresses at the manipulated object.

Other research efforts have been focused on handling multibodied objects, or even flexible objects [29.42–44]. Those objects are difficult to handle and, therefore, assembly of those objects in manufacturing industry is not automated. Also, cooperative control of multi-flexible-arm robots has been investigated [29.45, 46]. Once the modeling and control problem is solved (see Chap. 13), the flexible-arm robot is a robot with many merits [29.47]: it is lightweight, compliant, and hence safe, etc. Combining control methods for flexible-arm robots, such as vibration suppression, with the cooperative control methods presented in this chapter is straightforward [29.45]. Automated object-retrieval operation with a two-flexible-arm robot has been demonstrated in [29.46].

References

29.1 S. Fujii, S. Kurono: Coordinated computer control of a pair of manipulators, Proc. 4th IFToMM World Congress (Newcastle upon Tyne 1975) pp. 411–417

29.2 E. Nakano, S. Ozaki, T. Ishida, I. Kato: Cooperational control of the anthropomorphous manipulator 'MELARM', Proc. 4th Int. Symp. Ind. Robots (Tokyo 1974) pp. 251–260

29.3 K. Takase, H. Inoue, K. Sato, S. Hagiwara: The design of an articulated manipulator with torque control ability, Proc. 4th Int. Symp. Ind. Robots (Tokyo 1974) pp. 261–270

29.4 S. Kurono: Cooperative control of two artificial hands by a mini-computer, Prepr. 15th Joint Conf. on Automatic Control (1972) pp. 365–366, (in Japanese)

29.5 A.J. Koivo, G.A. Bekey: Report of workshop on coordinated multiple robot manipulators: planning, control, and applications, IEEE J. Robot. Autom. **4**(1), 91–93 (1988)

29.6 P. Dauchez, R. Zapata: Co-ordinated control of two cooperative manipulators: the use of a kinematic model, Proc. 15th Int. Symp. Ind. Robots (Tokyo 1985) pp. 641–648

29.7 N.H. McClamroch: Singular systems of differential equations as dynamic models for constrained robot systems, Proc. 1986 IEEE Int. Conf. on Robotics and Automation (San Francisco 1986) pp. 21–28

29.8 T.J. Tarn, A.K. Bejczy, X. Yun: New nonlinear control algorithms for multiple robot arms, IEEE Trans. Aerosp. Electron. Syst. **24**(5), 571–583 (1988)

29.9 S. Hayati: Hybrid position/force control of multi-arm cooperating robots, Proc. 1986 IEEE Int. Conf. on Robotics and Automation (San Francisco 1986) pp. 82–89

29.10 M. Uchiyama, N. Iwasawa, K. Hakomori: Hybrid position/force control for coordination of a two-arm robot, Proc. 1987 IEEE Int. Conf. on Robotics and Automation (Raleigh 1987) pp. 1242–1247

29.11 M. Uchiyama, P. Dauchez: A symmetric hybrid position/force control scheme for the coordination of two robots, Proc. 1988 IEEE Int. Conf. on Robotics and Automation (Philadelphia 1988) pp. 350–356

29.12 M. Uchiyama, P. Dauchez: Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots, Adv. Robot. **7**(4), 361–383 (1993)

29.13 I.D. Walker, R.A. Freeman, S.I. Marcus: Analysis of motion and internal force loading of objects grasped by multiple cooperating manipulators, Int. J. Robot. Res. **10**(4), 396–409 (1991)

29.14 R.G. Bonitz, T.C. Hsia: Force decomposition in cooperating manipulators using the theory of metric spaces and generalized inverses, Proc. 1994 IEEE Int. Conf. on Robotics and Automation (San Diego 1994) pp. 1521–1527

29.15 D. Williams, O. Khatib: The virtual linkage: a model for internal forces in multi-grasp manipulation, Proc. 1993 IEEE Int. Conf. on Robotics and Automation (Atlanta 1993) pp. 1025–1030

29.16 K.S. Sang, R. Holmberg, O. Khatib: The augmented object model: cooperative manipulation and parallel mechanisms dynmaics, Proceedings of the 2000 IEEE International Conference on Robotics and Automation (San Francisco 1995) pp. 470–475

29.17 J.T. Wen, K. Kreutz-Delgado: Motion and force control of multiple robotic manipulators, Automatica **28**(4), 729–743 (1992)

29.18 T. Yoshikawa, X.Z. Zheng: Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object, Int. J. Robot. Res. **12**, 219–230 (1993)

29.19 V. Perdereau, M. Drouin: Hybrid external control for two robot coordinated motion, Robotica **14**, 141–153 (1996)

29.20 H. Bruhm, J. Deisenroth, P. Schadler: On the design and simulation-based validation of an active compliance law for multi-arm robots, Robot. Auton. Syst. **5**, 307–321 (1989)

29.21 S.A. Schneider, R.H. Cannon Jr.: Object impedance control for cooperative manipulation: Theory and experimental results, IEEE Trans. Robot. Autom. **8**, 383–394 (1992)

29.22 R.G. Bonitz, T.C. Hsia: Internal force-based impedance control for cooperating manipulators, IEEE Trans. Robot. Autom. **12**, 78–89 (1996)

29.23 Y.-R. Hu, A.A. Goldenberg, C. Zhou: Motion and force control of coordinated robots during constrained motion tasks, Int. J. Robot. Res. **14**, 351–365 (1995)

29.24 Y.-H. Liu, S. Arimoto: Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperation, Int. J. Robot. Res. **17**, 232–247 (1998)

29.25 P. Chiacchio, S. Chiaverini, B. Siciliano: Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system, ASME J. Dyn. Syst. Meas. Contr. **118**, 691–697 (1996)

29.26 F. Caccavale, P. Chiacchio, S. Chiaverini: Task-Space regulation of cooperative manipulators, Automatica **36**, 879–887 (2000)

29.27 G.R. Luecke, K.W. Lai: A joint error-feedback approach to internal force regulation in cooperating manipulator systems, J. Robot. Syst. **14**, 631–648 (1997)

29.28 F. Caccavale, P. Chiacchio, S. Chiaverini: Stability analysis of a joint space control law for a two-manipulator system, IEEE Trans. Autom. Contr. **44**, 85–88 (1999)

29.29 P. Hsu: Coordinated control of multiple manipulator systems, IEEE Trans. Robot. Autom. **9**, 400–410 (1993)

29.30 P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano: Global task space manipulability ellipsoids for multiple arm systems, IEEE Trans. Robot. Autom. **7**, 678–685 (1991)

29.31 S. Lee: Dual redundant arm configuration optimization with task-oriented dual arm manipulability, IEEE Trans. Robot. Autom. **5**, 78–97 (1989)

29.32 T. Kokkinis, B. Paden: Kinetostatic performance limits of cooperating robot manipulators using force-velocity polytopes, Proc. of ASME Winter Annual Meeting–robotics Research (San Francisco 1989)

29.33 P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano: Task space dynamic analysis of multiarm system configurations, Int. J. Robot. Res. **10**, 708–715 (1991)

29.34 D.E. Orin, S.Y. Oh: Control of force distribution in robotic mechanisms containing closed kinematic chains, Trans. ASME J. Dyn. Syst. Meas. Contr. **102**, 134–141 (1981)

29.35 Y.F. Zheng, J.Y.S. Luh: Optimal load distribution for two industrial robots handling a single object, Proc. 1988 IEEE Int. Conf. on Robotics and Automation (Philadelphia 1988) pp. 344–349

29.36 I.D. Walker, S.I. Marcus, R.A. Freeman: Distribution of dynamic loads for multiple cooperating robot manipulators, J. Robot. Syst. **6**, 35–47 (1989)

29.37 M. Uchiyama: A unified approach to load sharing, motion decomposing, and force sensing of dual arm robots, Robotics Research: 5th Int. Symp., ed. by H. Miura, S. Arimoto (MIT, 1990) pp. 225–232

29.38 M.A. Unseren: A new technique for dynamic load distribution when two manipulators mutually lift a rigid object. Part 1: The proposed technique, Proc. First World Automation Congress (WAC '94), Vol. 2 (Maui 1994) pp. 359–365

29.39 M.A. Unseren: A new technique for dynamic load distribution when two manipulators mutually lift a rigid object. Part 2: Derivation of entire system model and control architecture, Proc. First World Automation Congress (WAC '94), Vol. 2 (Maui 1994) pp. 367–372

29.40 M. Uchiyama, T. Yamashita: Adaptive load sharing for hybrid controlled two cooperative manipulators, Proc. 1991 IEEE Int. Conf. on Robotics and Automation (Sacramento 1991) pp. 986–991

29.41 M. Uchiyama, Y. Kanamori: Quadratic programming for dextrous dual-arm manipulation. In: *Robotics, Mechatronics and Manufacturing Systems, Trans. IMACS/SICE Int. Symp. on Robotics, Mechatronics and Manufacturing Systems, Kobe, Japan, September 1992*, ed. by T. Takamori, K. Tsuchiya (Elsevier, North-Holland 1993) pp. 367–372

29.42 Y.F. Zheng, M.Z. Chen: Trajectory planning for two manipulators to deform flexible beams, Proc. 1993 IEEE Int. Conf. on Robotics and Automation (Atlanta 1993) pp. 1019–1024

29.43 M.M. Svinin, M. Uchiyama: Coordinated dynamic control of a system of manipulators coupled via a flexible object, Prepr. 4th IFAC Symp. on Robot Control (Capri 1994) pp. 1005–1010

29.44 T. Yukawa, M. Uchiyama, D.N. Nenchev, H. Inooka: Stability of control system in handling of a flexible object by rigid arm robots, Proc. 1996 IEEE Int. Conf. on Robotics and Automation (Minneapolis 1996) pp. 2332–2339

29.45 M. Yamano, J.-S. Kim, A. Konno, M. Uchiyama: Cooperative control of a 3D dual-flexible-arm robot, J. Intell. Robot. Syst. **39**, 1–15 (2004)

29.46 T. Miyabe, A. Konno, M. Uchiyama, M. Yamano: An approach toward an automated object retrieval operation with a two-arm flexible manipulator, Int. J. Robot. Res. **23**, 275–291 (2004)

29.47 M. Uchiyama, A. Konno: Modeling, controllability and vibration suppression of 3D flexible robots. In: *Robotics Research, The 7th Int. Symp*, ed. by G. Giralt, G. Hirzinger (Springer, London 1996) pp. 90–99

29.48 K. Munawar, M. Uchiyama: Slip compensated manipulation with cooperating multiple robots, 36th IEEE CDC (San Diego 1997)

29.49 D. Sun, J.K. Mills: Adaptive synchronized control for coordination of multirobot assembly tasks, IEEE Trans. Robot. Autom. **18**, 498–510 (2002)

29.50 A. Rodriguez-Angeles, H. Nijmeijer: Mutual synchronization of robots via estimated state feedback: a cooperative approach, IEEE Trans. Contr. Syst. Technol. **12**, 542–554 (2004)

29.51 K.-Y. Lian, C.-S. Chiu, P. Liu: Semi-decentralized adaptive fuzzy control for cooperative multirobot systems with H-inf motion/internal force tracking performance, IEEE Trans. Syst. Man Cybern. – Part B: Cybernetics **32**, 269–280 (2002)

29.52 W. Gueaieb, F. Karray, S. Al-Sharhan: A robust adaptive fuzzy position/force control scheme for cooperative manipulators, IEEE Trans. Contr. Syst. Technol. **11**, 516–528 (2003)

29.53 J. Gudiño-Lau, M.A. Arteaga, L.A. Muñoz, V. Parra-Vega: On the control of cooperative robots without velocity measurements, IEEE Trans. Contr. Syst. Technol. **12**, 600–608 (2004)

29.54 H. Inoue: Computer controlled bilateral manipulator, Bull. JSME **14**(69), 199–207 (1971)

29.55 M. Uchiyama, T. Kitano, Y. Tanno, K. Miyawaki: Cooperative multiple robots to be applied to industries, Proc. World Automation Congress (WAC '96), Vol. 3 (Montpellier 1996) pp. 759–764
29.56 B.M. Braun, G.P. Starr, J.E. Wood, R. Lumia: A framework for implementing cooperative motion on industrial controllers, IEEE Trans. Robot. Autom. **20**, 583–589 (2004)
29.57 D. Sun, J.K. Mills: Manipulating rigid payloads with multiple robots using compliant grippers, IEEE/ASME Trans. Mechatron. **7**, 23–34 (2002)
29.58 J.Y.S. Luh, Y.F. Zheng: Constrained relations between two coordinated industrial robots for motion control, Int. J. Robot. Res. **6**, 60–70 (1987)
29.59 A.J. Koivo, M.A. Unseren: Reduced order model and decoupled control architecture for two manipulators holding a rigid object, ASME J. Dyn. Syst. Meas. Contr. **113**, 646–654 (1991)
29.60 M.A. Unseren: Rigid body dynamics and decoupled control architecture for two strongly interacting manipulators manipulators, Robotica **9**, 421–430 (1991)
29.61 J. Duffy: The fallacy of modern hybrid control theory that is based on "Orthogonal Complements" of twist and wrench spaces, J. Robot. Syst. **7**, 139–144 (1990)
29.62 K.L. Doty, C. Melchiorri, C. Bonivento: A theory of generalized inverses applied to robotics, Int. J. Robot. Res. **12**, 1–19 (1993)
29.63 O. Khatib: Object manipulation in a multi-effector robot system,. In: *Robotics Research*, Vol. 4, ed. by R. Bolles, B. Roth (MIT Press, Cambridge 1988) pp. 137–144
29.64 O. Khatib: Inertial properties in robotic manipulation: An object level framework, Int. J. Robot. Res. **13**, 19–36 (1995)
29.65 F. Caccavale, L. Villani: Impedance control of cooperative manipulators, Mach. Intell. Robot. Contr. **2**, 51–57 (2000)