Asymmetric Dual-Arm Task Execution Using an Extended Relative Jacobian

Citation for the original published paper (version of record):
Almeida, D., Karayiannidis, Y. (2022). Asymmetric Dual-Arm Task Execution Using an Extended Relative Jacobian. Springer Proceedings in Advanced Robotics, 20 SPAR: 18-34.
http://dx.doi.org/10.1007/978-3-030-95459-8_2

(article starts on next page)

Asymmetric Dual-Arm Task Execution Using an Extended Relative Jacobian

Diogo Almeida[1](✉) and Yiannis Karayiannidis[2]

[1] Division of Robotics, Perception and Learning, KTH Royal Institute of Technology,
100 44 Stockholm, Sweden
diogoa@kth.se

[2] Department of Electrical Engineering, Chalmers University of Technology,
412 96 Gothenburg, Sweden
yiannis@chalmers.se

Abstract. Coordinated dual-arm manipulation tasks can be broadly characterized as possessing absolute and relative motion components. Relative motion tasks, in particular, are inherently redundant in the way they can be distributed between end-effectors. In this work, we analyse cooperative manipulation in terms of the asymmetric resolution of relative motion tasks. We discuss how existing approaches enable the asymmetric execution of a relative motion task, and show how an asymmetric relative motion space can be defined. We leverage this result to propose an extended relative Jacobian to model the cooperative system, which allows a user to set a concrete degree of asymmetry in the task execution. This is achieved without the need for prescribing an absolute motion target. Instead, the absolute motion remains available as a functional redundancy to the system. We illustrate the properties of our proposed Jacobian through numerical simulations of a novel differential Inverse Kinematics algorithm.

1 Introduction

Consider a dual-armed robotic system, tasked with executing a *coordinated* manipulation task. As opposed to *uncoordinated* tasks, coordinated manipulation benefits from treating the two arms in the robotic system as a single kinematic chain. Concretely, the task space for coordinated manipulation can be defined in terms of *absolute* and *relative* motion components. Absolute motion is equivalent to the resultant motion from external forces applied to a jointly carried object. Relative motion, conversely, refers to the motion of one end-effector w.r.t the other [27,28]. This type of motion can be used to model tasks such as,

This work is partially supported by the Swedish Foundation for Strategic Research project GMT14-0082 FACT.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-030-95459-8_2.

T. Asfour et al. (Eds.): ISRR 2019, SPAR 20, pp. 18–34, 2022.
https://doi.org/10.1007/978-3-030-95459-8_2

e.g., assembly or tool manipulation. The formulation of the manipulation task space in terms of absolute and relative motion is called the Cooperative Task Space (CTS) [8].

We are interested in analysing the resolution of the CTS variables into end-effector velocities. Recent work proposes an Extended CTS (ECTS) definition which makes use of a novel definition of absolute motion to attain an *asymmetric* resolution of the relative motion variable [24,25]. Alternatively, it is common to model solely the relative motion as a primary task quantity, by using the relative Jacobian formulation [17] when solving the dual-armed system's differential Inverse Kinematics (IK). In many cases, a secondary task is added which constrains the absolute motion of the system resulting in asymmetric relative motion [1,11–13].

In the following text, we discuss the CTS formulations and observe how a conflict between the definition of absolute and relative motion tasks is introduced by the ECTS method, Sect. 3. In fact, ECTS relies on an asymmetric absolute motion definition, which we argue is at odds with the concept of absolute motion: commanding an 'asymmetric absolute motion' will result in the appearance of relative motion components. If we consider an absolute motion task such as jointly carrying an object, this will inevitably lead to internal forces on the object. On the other hand, relative motion tasks are *inherently redundant*. The resolution of the relative motion task into the robot end-effectors can be arbitrary without any fundamental conflict with the concept of relative motion [3].

We leverage our observations to propose an asymmetric relative motion space, which enables the asymmetric resolution of the relative motion without resorting to a redefinition of the absolute motion space, Sect. 4. The new space leads to the proposal of a novel relative Jacobian formulation and a corresponding differential IK algorithm, Sect. 5. This allows a user to specify a concrete degree of asymmetry in the relative task resolution, within a relative Jacobian framework. We illustrate the key properties of our approach through numerical simulations, Sect. 6, and make our code freely available, which, in addition to the simulations, implements velocity controllers that employ all of the discussed methods[1].

2 Related Work

A common solution to the problem of bimanual cooperative manipulation is to employ a completely asymmetrical leader-follower (or master-slave) approach [2,4,18,29,30]. Alternatively, by considering the external and internal forces on a jointly held rigid object, the task can be modelled in terms of absolute and relative motion components [27,28]. The CTS definition results from a modelling approach which is independent of the statics of the dual-armed system [7,8]. The ECTS [24,25] is obtained by redefining the absolute motion space of the coordinated system. CTS-based approaches have been used to describe coordinated tasks in, e.g., human-robot interaction settings [20], the cooperative manipulation of a mechanism [3] or the execution of a bimanual dexterous manipulation task [9].

[1] https://github.com/diogoalmeida/asymmetric_manipulation.

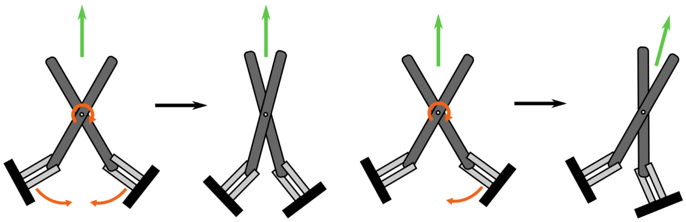

(a) **Symmetric** execution. (b) **Asymmetric** execution

Fig. 1. A simple relative motion task is to open/close a pair of scissors. Note that an asymmetric execution results in a change of the average orientation (green arrow) of the two scissors' pieces.

Relative Jacobian methods model solely the relative motion task [13,16,17], making it a common choice when addressing tasks that require just the relative motion space, such as machining [14,22,23], assembly [1] or drawing [15]. The absolute motion is part of the relative Jacobian's redundant space. This can be exploited, e.g., to enhance self-collision and obstacle avoidance as secondary tasks [19], or for joint-limit avoidance [11,12,21].

Hierarchical quadratic programs (HQP) [10] are an alternative approach to obtain solutions to the problem of inverse differential kinematics. While in this article we focus on pseudo-inverse solutions to the differential IK problem, all the discussed Jacobian formulations can be employed in the context of HQP, as showed in, e.g., [6] and [26] for the relative Jacobian.

3 Cooperative Motion Spaces

In this section, we perform a task-space analysis of existing cooperation strategies. From the kinematic point of view, we will consider master-slave methods as examples of purely asymmetric task execution, which, as we will show, are covered by the analysed strategies.

CTS methods employ the full dual-arm task space, by defining an *absolute* and a *relative* motion spaces. A possible approach to obtain *asymmetric* relative motion is to redefine the absolute motion space, as is the case with ECTS. We observe that this approach introduces conflicts between the relative and absolute motion tasks.

3.1 Notation

Consider a dual-armed system composed by two robotic manipulators. Let $\{h_i\}$ denote a generic coordinate frame, e.g., of the i-th manipulator's end-effector, where $i \in \{1, 2\}$. Each frame is defined by a position, $\mathbf{p}_i \in \mathbb{R}^3$ and orientation $\mathbf{R}_i \in \mathcal{SO}(3)$, expressed in a common base frame. We write the angle-axis representation of $\mathbf{R}_i$ as $\mathbf{R}_{\mathbf{k}}(\vartheta_i)$, where ϑ_i is the angle one must rotate about the axis $\mathbf{k}$ to obtain $\mathbf{R}_i$. The twist at each end-effector is defined as $\mathbf{v}_i = [\dot{\mathbf{p}}_i^\top \ \boldsymbol{\omega}_i^\top]^\top$, where $\boldsymbol{\omega}_i \in \mathbb{R}^3$ denotes the end-effectors' angular velocity. Finally, we denote the nullspace of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ as $\mathcal{N}(\mathbf{A})$.

3.2 Absolute and Relative Motion Spaces

The absolute and relative motion components can be derived from properly defined motion frames.

Cooperative Task Space. In CTS [8], absolute and relative motion frames are defined, respectively $\{h_a\}$ and $\{h_r\}$, such that

$$\mathbf{p}_a = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2) \quad \text{and} \quad \mathbf{R}_a = \mathbf{R}_1 \mathbf{R}_{\mathbf{k}_{1,2}}\left(\frac{\vartheta_{1,2}}{2}\right), \tag{1}$$

where $\mathbf{k}_{1,2}$ and $\vartheta_{1,2}$ are extracted from the angle-axis representation of ${}^1\mathbf{R}_2 = \mathbf{R}_1^\top \mathbf{R}_2$, and

$$\mathbf{p}_r = \mathbf{p}_2 - \mathbf{p}_1, \quad \mathbf{R}_r = {}^1\mathbf{R}_2. \tag{2}$$

The absolute and relative motion twists, respectively $\mathbf{v}_a$ and $\mathbf{v}_r$ can be obtained through differentiation,

$$\mathbf{v}_a = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2) \quad \text{and} \quad \mathbf{v}_r = \mathbf{v}_2 - \mathbf{v}_1. \tag{3}$$

In task space, this relation can be expressed by a *linking* matrix, $\mathbf{L}_{cts} \in \mathbb{R}^{12}$. Let $\mathbf{v}_{cts} = [\mathbf{v}_a^\top \ \mathbf{v}_r^\top]^\top$ and $\mathbf{v} = [\mathbf{v}_1^\top \ \mathbf{v}_2^\top]^\top$. Then,

$$\mathbf{v}_{cts} = \mathbf{L}_{cts}\mathbf{v}, \quad \mathbf{L}_{cts} = \begin{bmatrix} \frac{1}{2}\mathbf{I}_6 & \frac{1}{2}\mathbf{I}_6 \\ -\mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix}. \tag{4}$$

The CTS linking matrix $\mathbf{L}_{cts}$ is square and nonsingular, and $\mathbf{v}$ can be recovered through matrix inversion,

$$\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_6 & -\frac{1}{2}\mathbf{I}_6 \\ \mathbf{I}_6 & \frac{1}{2}\mathbf{I}_6 \end{bmatrix} \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_r \end{bmatrix}. \tag{5}$$

It is clear from (5) that the coordinated task is divided *symmetrically* between the two end-effectors: each end-effector executes the prescribed absolute motion, and the relative motion task is divided evenly among them.

Extended Cooperative Task Space. The CTS formulation has been extended in [24,25]. A different definition for the absolute frame is adopted,

$$\mathbf{p}_a = \alpha\mathbf{p}_1 + (1-\alpha)\mathbf{p}_2, \qquad \mathbf{R}_a = \mathbf{R}_1 \mathbf{R}_{\mathbf{k}_{1,2}}\left((1-\alpha)\vartheta_{1,2}\right), \tag{6}$$

with $0 \leq \alpha \leq 1$. The ECTS linking matrix is given by

$$\mathbf{L}_E(\alpha) = \begin{bmatrix} \alpha\mathbf{I}_6 & (1-\alpha)\mathbf{I}_6 \\ -\mathbf{I}_6 & \mathbf{I}_6, \end{bmatrix} \tag{7}$$

and $\mathbf{v}_{cts} = \mathbf{L}_E(\alpha)\mathbf{v}$, as in (4). The effect of the cooperation parameter α is clear through the inversion of (7),

$$\mathbf{L}_E(\alpha)^{-1} = \begin{bmatrix} \mathbf{I}_6 & -(1-\alpha)\mathbf{I}_6 \\ \mathbf{I}_6 & \alpha\mathbf{I}_6 \end{bmatrix}, \tag{8}$$

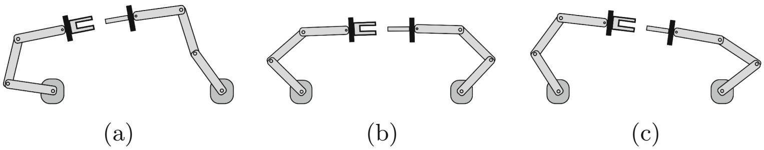


Fig. 2. The absolute motion of the cooperative system is redundant w.r.t the relative motion task.

That is, by setting α, asymmetric relative motion can be achieved, such that $\mathbf{v}_a = \mathbf{0}$ in the new absolute frame. The relative motion task can be solved in a serial (master-slave, $\alpha = 0$ or $\alpha = 1$), blended (asymmetrical, $\alpha \neq 0.5$) or parallel (symmetrical, $\alpha = 0.5$) mode of cooperation. Note that the introduction of asymmetries in the relative motion affects the symmetric absolute pose, Fig. 1.

Both $\mathbf{L}_{cts}$ and $\mathbf{L}_E(\alpha)$ are composed by an absolute and a relative part,

$$\mathbf{L}_{cts} = \begin{bmatrix} \mathbf{L}_a \\ \mathbf{L}_r \end{bmatrix} \quad \mathbf{L}_E(\alpha) = \begin{bmatrix} \boldsymbol{\mathcal{L}}_a(\alpha) \\ \mathbf{L}_r \end{bmatrix}, \tag{9}$$

where $\mathbf{L}_a, \mathbf{L}_r, \boldsymbol{\mathcal{L}}_a(\alpha) \in \mathbb{R}^{6\times 12}$. These matrices depend on the definition of the frames where the absolute and relative motion are expressed. For both CTS and ECTS, the relative motion is given by $\mathbf{v}_r = \mathbf{L}_r\mathbf{v}$. CTS adopts the symmetric resolution of the absolute motion, $\mathbf{v}_a = \mathbf{L}_a\mathbf{v}$ while ECTS makes use of an asymmetric formulation, $\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\mathbf{v}$.

The (E)CTS motion space fully specifies the cooperative motion in terms of absolute and relative variables. In many tasks, however, this overconstrains the problem. For example, a peg-in-hole assembly can be executed by having each robot arm grasp a part and executing a relative motion between its end-effectors. The absolute motion is a functional redundancy in this case, Fig. 2. Other examples include machining [23] or drawing [15].

Relative Motion Space. For tasks which can be solved exclusively through relative motion, it is possible to specify a desired $\mathbf{v}_r$ only. This removes the need to adopt the 12×12 dimensional linking matrices in (9). In this scenario, we can use the Moore-Penrose pseudo-inverse, $\mathbf{L}_r^\dagger = \mathbf{L}_r^\top(\mathbf{L}_r\mathbf{L}_r^\top)^{-1}$, to resolve a desired relative motion into velocities of each robot end-effector,

$$\mathbf{v} = \mathbf{L}_r^\dagger\mathbf{v}_r = \frac{1}{2}\begin{bmatrix} -\mathbf{I}_6 \\ \mathbf{I}_6 \end{bmatrix}\mathbf{v}_r, \tag{10}$$

which matches the relative part of the CTS solution (5). An homogenous solution can be added through a nullspace projection, e.g.,

$$\begin{aligned} \mathbf{v} &= \mathbf{L}_r^\dagger\mathbf{v}_r + (\mathbf{I}_{12} - \mathbf{L}_r^\dagger\mathbf{L}_r)[\mathbf{I}_6 \quad \mathbf{0}_6]^\dagger\mathbf{v}_{1_d} \\ &= \frac{1}{2}\begin{bmatrix} \mathbf{v}_{1_d} - \mathbf{v}_r \\ \mathbf{v}_{1_d} + \mathbf{v}_r \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \mathbf{I}_6 & -\mathbf{I}_6 \\ \mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix}\begin{bmatrix} \mathbf{v}_{1_d} \\ \mathbf{v}_r \end{bmatrix}. \end{aligned} \tag{11}$$

In this solution, a secondary task $\{h_1\}$ is being mapped to the absolute motion space, with $\mathbf{v}_a = \frac{1}{2}\mathbf{v}_{1_d}$. The resulting velocity distribution is analogous to the

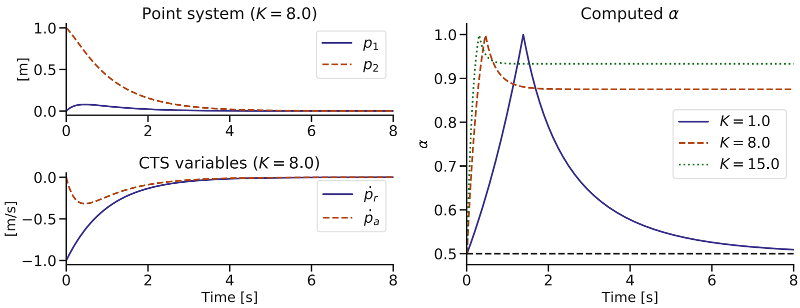


Fig. 3. Numerical simulation of the point-system (12). Different values of K affect the degree of asymmetry α in the execution of the relative motion task.

CTS case, Eq. (5). However, if the secondary objective is not prescribed in $\{h_a\}$, a conflict between primary and secondary tasks is introduced which will induce asymmetries in the motion of the system's end-effectors.

Example 1 (Constructing asymmetric relative motion). Let $p_1, p_2 \in \mathbb{R}$, with initial state $p_1(0) = 0$ and $p_2(0) = 1$, and consider the relative velocity command $\dot{p}_{r_d} = p_1 - p_2$. Additionally, let $\dot{p}_{1_d} = -Kp_1$, with $K > 0$. The solution (11), applied to this one-dimensional case, yields the autonomous system

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -(K+1) & 1 \\ 1-K & -1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}. \tag{12}$$

The relative velocity of this system is given by $\dot{p}_r = \dot{p}_2 - \dot{p}_1 = \dot{p}_{r_d}$, i.e., the desired relative command is attained. The degree of asymmetry in the motion of the two points can be computed through $\alpha = \frac{|\dot{p}_2|}{|\dot{p}_1|+|\dot{p}_2|}$ and will depend on the secondary task gain K. We depict the time evolution of system (12), as well as α for different values of K in Fig. 3. It is clear that for a larger K the more asymmetric is the motion of the two points, since the induced absolute motion will regulate p_1 to remain at the origin. For $K = 8$ in particular, it can be seen that the injected absolute motion $\dot{p}_a$ allows p_2 to contribute the most to the relative motion task.

3.3 Asymmetric Resolution of the CTS Variables

We have seen in Sect. 3.2 that inverting the mapping (4) yields a symmetric distribution of the CTS variables into the two robot end-effectors. In the current literature, two predominant coordination approaches achieve an asymmetric execution of the CTS variables:

1. In the ECTS [24,25], a redefined absolute motion space constrains the resolution of the relative motion to be asymmetric, Sect. 3.2.
2. Methods which consider only the relative motion space often add a secondary task to one of the end-effectors [1,11–13], which results in asymmetric behavior, see Example 1.

Both methods combine the CTS variables to achieve asymmetric behavior. The ECTS solution for the absolute motion introduces relative motion components,

$$\mathbf{v}_r = \mathbf{L}_r \boldsymbol{\mathcal{L}}_a(\alpha)^\dagger \mathbf{v}_a = \frac{1-2\alpha}{\alpha^2 + (1-\alpha)^2}\mathbf{v}_a, \tag{13}$$

while the nullspace projection (11) maps the secondary task of the system to the absolute motion space. We will show how this projection can be modified to introduce a deliberate degree of asymmetry in the execution of the relative motion task.

4 Defining an Asymmetric Relative Motion Space

Consider that we wish to control the relative motion between the end-effectors in an *asymmetric* manner. Analogously to the ECTS approach, we would like to be able to specify a degree of cooperation between the arms. This can be achieved through a redefined relative motion frame. If we denote the angle-axis representations of $\mathbf{R}_1 = \mathbf{R}_{\mathbf{k}_1}(\vartheta_1)$ and $\mathbf{R}_2 = \mathbf{R}_{\mathbf{k}_2}(\vartheta_2)$, then the asymmetric relative motion frame $\{h_r\}$ can be defined as

$$\begin{aligned}\mathbf{p}_r &= \frac{\alpha\mathbf{p}_2 - (1-\alpha)\mathbf{p}_1}{(1-\alpha)^2 + \alpha^2}\\ \mathbf{R}_r &= \mathbf{R}_{\mathbf{k}_1}^\top\left(\frac{(1-\alpha)\vartheta_1}{(1-\alpha)^2+\alpha^2}\right)\mathbf{R}_{\mathbf{k}_2}\left(\frac{\alpha\vartheta_2}{(1-\alpha)^2+\alpha^2}\right).\end{aligned} \tag{14}$$

Asymmetric relative motion can be obtained through the differentiation of (14),

$$\mathbf{v}_r = \boldsymbol{\mathcal{L}}_r(\alpha)\mathbf{v}, \tag{15}$$

where the asymmetric relative linking matrix $\boldsymbol{\mathcal{L}}_r(\alpha)$ is

$$\boldsymbol{\mathcal{L}}_r(\alpha) = \frac{1}{(1-\alpha)^2+\alpha^2}\begin{bmatrix}-(1-\alpha)\mathbf{I}_6 & \alpha\mathbf{I}_6\end{bmatrix}. \tag{16}$$

The particular solution to (15) is analogous to (10) and corresponds to the ECTS asymmetric distribution of the relative motion (8),

$$\mathbf{v} = \boldsymbol{\mathcal{L}}_r(\alpha)^\dagger\mathbf{v}_r = \begin{bmatrix}-(1-\alpha)\mathbf{I}_6\\ \alpha\mathbf{I}_6\end{bmatrix}\mathbf{v}_r. \tag{17}$$

Theorem 1. *The motion space defined by $\{h_r\}$ in (14) is characterized by the following properties:*

1. *Commanding a relative velocity in $\{h_r\}$, Eq. (17), renders the asymmetric absolute motion space (6) invariant.*
2. *The inverse mapping $\boldsymbol{\mathcal{L}}_r(\alpha)^\dagger$ is a generalized inverse of $\mathbf{L}_r$.*

Proof. Let

$$\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\mathbf{v}, \tag{18}$$

and consider the solution (17)*. We have,*

$$\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\boldsymbol{\mathcal{L}}_r(\alpha)^{\dagger}\mathbf{v}_r = \mathbf{0}, \tag{19}$$

and thus, the asymmetric absolute motion frame (6) *remains invariant to velocities commanded in the motion frame from* (14)*. $\boldsymbol{\mathcal{L}}_r(\alpha)$ acts as a generalized inverse to $\mathbf{L}_r$ since*

$$\mathbf{L}_r\boldsymbol{\mathcal{L}}_r(\alpha)^{\dagger} = \begin{bmatrix}-\mathbf{I}_6 \; \mathbf{I}_6\end{bmatrix}\begin{bmatrix}-(1-\alpha)\mathbf{I}_6 \\ \alpha\mathbf{I}_6\end{bmatrix} = \mathbf{I}_6. \tag{20}$$

Corollary 1. *The solution for the asymmetric relative motion space* (17) *can be obtained by adding an homogeneous component to* (10)*.*

Proof. This follows straightforwardly from the fact that $\boldsymbol{\mathcal{L}}_r(\alpha)$ acts as a generalized inverse to $\mathbf{L}_r$ (20)*,*

$$\mathbf{v} = \mathbf{L}_r^{\dagger}\mathbf{v}_r + (\mathbf{I}_{12} - \mathbf{L}_r^{\dagger}\mathbf{L}_r)\boldsymbol{\mathcal{L}}_r(\alpha)^{\dagger}\mathbf{v}_r = \boldsymbol{\mathcal{L}}_r(\alpha)^{\dagger}\mathbf{v}_r. \tag{21}$$

The solution (21) generates the required absolute motion to ensure the degree of asymmetry α in the execution of the prescribed relative velocity $\mathbf{v}_r$,

$$\mathbf{v}_a = \mathbf{L}_a\boldsymbol{\mathcal{L}}_r(\alpha)^{\dagger}\mathbf{v}_r = \frac{2\alpha-1}{2}\mathbf{v}_r. \tag{22}$$

We can use this result to construct a relative Jacobian method for solving the system's differential IK which takes into account a desired degree of asymmetry.

5 Differential Inverse Kinematics

The prescribed cooperative motion can be distributed between the two end-effectors, as seen in the previous section, and each arm can solve its differential IK separately. Alternatively, we can derive differential IK algorithms which directly resolve a desired cooperative motion to the joint state of the complete dual-arm chain.

5.1 Notation

Let $\mathbf{q}_i \in \mathbb{R}^n$ represent the joint variables of the i-th manipulator, with $n \geq 6$, and $\mathbf{J}_i(\mathbf{q}_i) \in \mathbb{R}^{6\times n}$ its Jacobian, such that $\mathbf{v}_i = \mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i$. We will omit the Jacobian's dependency on the joint variables for the remaining of this text, and assume that the manipulators' Jacobians are full rank, i.e., the manipulators are not operating in a singular configuration. Finally, let $\mathbf{S}(\mathbf{a}) \in \mathbb{R}^{3\times 3}$ be the skew-symmetric matrix such that, for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

5.2 Joint Jacobians

Let $\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0}_{6\times n} \\ \mathbf{0}_{6\times n} & \mathbf{J}_2 \end{bmatrix}$. We can solve the cooperative task by using any of the methods in Sect. 3 and computing the differential IK for each individual manipulator. As an example, a relative motion task can be solved into joint space by computing

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\mathbf{L}_r^{\dagger}\mathbf{v}_r, \tag{23}$$

where $\mathbf{q} = [\mathbf{q}_1^{\top},\ \mathbf{q}_2^{\top}]^{\top}$. The nullspace of $\mathbf{J}$ has dimension $\dim(\mathcal{N}(\mathbf{J})) = 2n - 12$. Alternatively, all the previously discussed cooperative motion definitions can be extended to a mapping from task to joint spaces through the construction of joint Jacobians. This is equivalent to treating the two manipulators as a single kinematic chain. We can obtain the joint Jacobians through a pre-multiplication with the appropriate linking matrix, e.g., $\mathbf{J}_{cts} = \mathbf{L}_{cts}\mathbf{J} \in \mathbb{R}^{12\times 2n}$. It is often convenient, however, to express the cooperative task w.r.t object frames rigidly connected to each end-effector, $\{h_{o_i}\}$. These can represent, e.g., the tip of the peg and the center of the hole in a peg-in-hole type of assembly task. In this case, a transformation between the twists at $\{h_{o_i}\}$ and the corresponding $\{h_i\}$ is needed.

We define two virtual sticks, which connect $\mathbf{p}_{o_i}$ to the end-effector's positions $\mathbf{p}_i$, such that $\mathbf{r}_i = \mathbf{p}_{o_i} - \mathbf{p}_i$. The necessary screw transformation is defined as $\mathbf{W}_i = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\mathbf{r}_i) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix}$, such that $\mathbf{v}_{o_i} = \mathbf{W}_i\mathbf{v}_i$. Let $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{W}_2 \end{bmatrix}$. The joint Jacobians are

$$\mathbf{J}_{cts} = \mathbf{L}_{cts}\mathbf{W}\mathbf{J} \qquad \mathbf{J}_E(\alpha) = \mathbf{L}_E(\alpha)\mathbf{W}\mathbf{J}, \tag{24}$$

for the CTS formulations, where $\mathbf{J}_{cts},\ \mathbf{J}_E(\alpha) \in \mathbb{R}^{12\times 2n}$ and

$$\mathbf{J}_r = \mathbf{L}_r\mathbf{W}\mathbf{J}, \tag{25}$$

in case only a relative motion task is specified, with $\mathbf{J}_r \in \mathbb{R}^{6\times 2n}$. Note that, in general, $\mathbf{J}_r^{\dagger} \neq \mathbf{J}^{\dagger}\mathbf{W}^{\dagger}\mathbf{L}_r^{\dagger}$. In fact, treating the dual-armed system as a single kinematic chain results, in general, in a larger dimension of the Jacobian nullspace, as $\operatorname{rank}(\mathbf{J}_r) \leq 6$ and thus

$$\dim(\mathcal{N}(\mathbf{J}_r)) \geq \dim(\mathcal{N}(\mathbf{J})) = \dim(\mathcal{N}(\mathbf{J}_1)) + \dim(\mathcal{N}(\mathbf{J}_2)). \tag{26}$$

In practice, the larger nullspace includes absolute motion components which are not present when solving (23). The Jacobian in (25) is often called relative Jacobian.

The forward differential kinematics for a relative motion task are expressed as

$$\mathbf{v}_r = \mathbf{J}_r\dot{\mathbf{q}}. \tag{27}$$

In the following, we will focus on finding feasible joint space solutions for (27). It is well known that such solutions have the general form

$$\dot{\mathbf{q}} = \mathbf{J}_r^{\dagger}\mathbf{v}_r + (\mathbf{I}_{2n} - \mathbf{J}_r^{\dagger}\mathbf{J}_r)\boldsymbol{\zeta}, \tag{28}$$

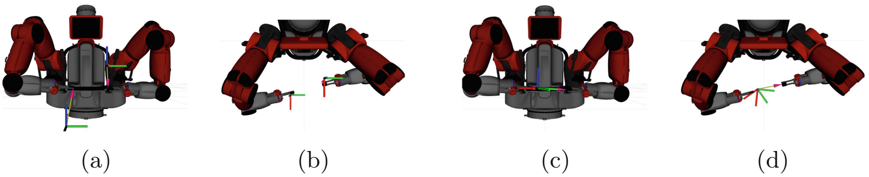


Fig. 4. Initial robot configurations for the case studies. Figure 4a and 4b: linear displacement between frames seen from a front and a top view. Figure 4c and 4d: rotational displacement seen from a front and a top view.

with $(\mathbf{I}_{2n} - \mathbf{J}_r^{\dagger}\mathbf{J}_r)\boldsymbol{\zeta}$ being part of the homogeneous solution to (27). The joint space vector $\boldsymbol{\zeta} \in \mathbb{R}^{2n}$ composes a secondary task, added to $\mathbf{v}_r$ in a task-priority manner. By setting $\boldsymbol{\zeta} = \mathbf{0}$ we get the minimum norm solution for $\dot{\mathbf{q}}$. The symmetric absolute motion space is orthogonal to the relative motion space used in the relative Jacobian definition (25),

$$\mathbf{L}_r\mathbf{L}_a^{\dagger} = \mathbf{0}. \tag{29}$$

This absolute motion is then a functional redundancy to (27) and thus, in general, its minimum norm solution can contain absolute motion components, depending on the manipulators' design. The desirability of this property depends on the task requirements. If the absolute motion must be prescribed, using the CTS is ideal. In light of the discussion in Sect. 3.3, we argue that the ECTS can be used to achieve asymmetric relative motion when $\mathbf{v}_a = \mathbf{0}$. However, if $\mathbf{v}_a \neq 0$ is to be commanded, ECTS requires $\alpha = 0.5$[2]. Otherwise, the resulting asymmetric absolute motion command will contain relative motion components, which is undesirable. Alternatively, using (23) prevents any non-specified absolute motion from occurring. When it is acceptable for absolute motion to be exploited as a functional redundancy, exploitation strategies include, e.g., workspace obstacle and self-collisions avoidance [19].

5.3 Asymmetric Relative Jacobian

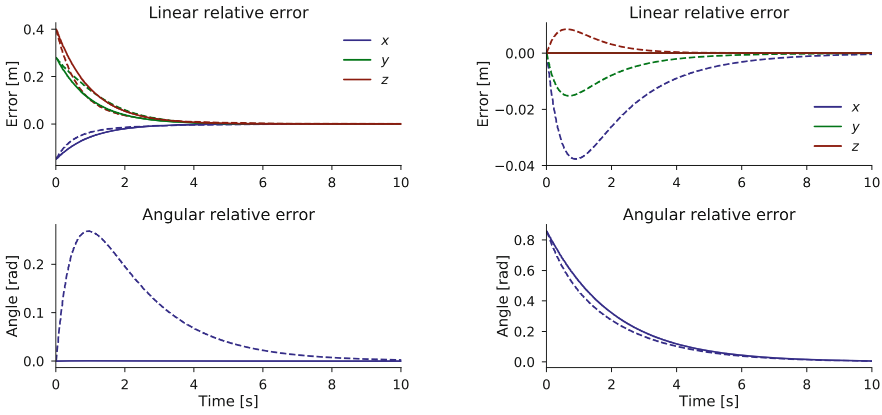
A common choice for $\boldsymbol{\zeta}$ is to specify a motion for one of the system's end-effectors [1,11–13], e.g.,

$$\boldsymbol{\zeta} = [\mathbf{J}_1 \ \mathbf{0}]^{\dagger}\mathbf{v}_1. \tag{30}$$

As seen in Example 1, this secondary task will be mapped into the system's absolute motion space and induce asymmetries in the execution of $\mathbf{v}_r$. We propose instead to use the asymmetric relative motion space defined in Sect. 4 to extend the relative Jacobian methods and allow for setting the degree of cooperation through a parameter. We use (16) to define an asymmetric Jacobian,

$$\mathbf{J}_r(\alpha) = \boldsymbol{\mathcal{L}}_r(\alpha)\mathbf{W}\mathbf{J}. \tag{31}$$

[2] In fact, this is the case in [25].



(a) **Translational** relative motion task. (b) **Rotational** relative motion task.

Fig. 5. Evolution of the relative errors in the two types of motion task, when $\alpha = 0.8$. The dashed lines represent the solution when using $\boldsymbol{\zeta}$ from (32) as a primary task. The solid lines use (32) in a task priority manner (28).

We can now solve the differential IK by defining

$$\boldsymbol{\zeta} = \mathbf{J}_r(\alpha)^{\dagger}\mathbf{v}_r. \tag{32}$$

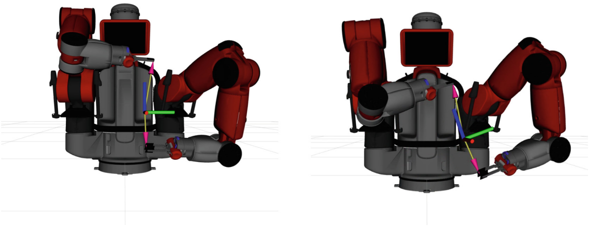
Imposing (32) as a secondary task will filter out undesired asymmetric absolute motion, which can occur as part of the minimum norm solution represented by the pseudoinverse, since it belongs to an orthogonal motion space, Eq. (19). The desired task space relative motion will be preserved, however, as shown in Corollary 1, and as we will illustrate in Sect. 6. Note that asymmetric absolute motion necessarily contains relative motion components, which is in general undesirable: for many tasks, non-prescribed relative motion can lead to internal stresses on the robot manipulator.

5.4 Relative Task

The choice of $\mathbf{v}_r$ in (28) defines the relative motion task. This can be set as the output of a task-specific controller [3], a feedforward command from a teleoperator, or as an error signal from a pose regulation task [5,25]. In our examples, we will assume without loss of generality that the task is to align the coordinate frames $\{h_{o_i}\}$. The alignment error is given by $\tilde{\mathbf{p}} = \mathbf{p}_{o_2} - \mathbf{p}_{o_1}$ for the displacement between frames and $\tilde{\mathbf{R}} = \mathbf{R}_{o_1}^{\top}\mathbf{R}_{o_2}$ for the orientation. If we denote the error quaternion as $\tilde{\mathcal{Q}} = (\tilde{\boldsymbol{\xi}}, w)$, where $\tilde{\boldsymbol{\xi}}$ is the vector and w the scalar part, we set $\mathbf{v}_r$ as the feedback control law [5],

$$\mathbf{v}_r = -\mathbf{K}_p \begin{bmatrix} \tilde{\mathbf{p}} \\ \mathbf{R}_{o_1}\tilde{\boldsymbol{\xi}} \end{bmatrix}, \tag{33}$$

where $\mathbf{K}_p \in \mathbb{R}^{6\times 6}$ is positive definite.



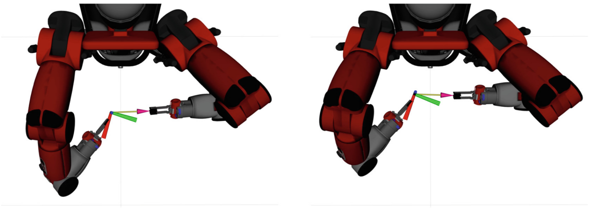
(a) ECTS (b) Proposed

Fig. 6. Results for the execution of a pure **translational** task.

6 Case Studies

We illustrate different properties of the novel relative Jacobian (31) with two case studies, in which the primary task is a relative motion task given by (33) with $\mathbf{K}_p = \mathbf{I}_6$. The case studies are based on a simulated Rethink Robotics' Baxter dual-armed robot. Two relative motion tasks are considered, one where the alignment error is purely translational, Fig. 4a–4b, the other where the error is purely rotational, Fig. 4c–4d. The initial poses of the object frames, expressed in the reference frame (Baxter's torso frame), are given by $\mathbf{p}_{o_1} = [0.36, 0.15, 0.36]^\top$ and $\mathbf{p}_{o_2} = [0.45, 0.0, 0.21]^\top$, with $\mathbf{R}_{o_1} = \mathbf{R}_{o_2} = \mathbf{I}_3$. The subscript index $i = 1$ corresponds to the left and $i = 2$ to the right manipulator.

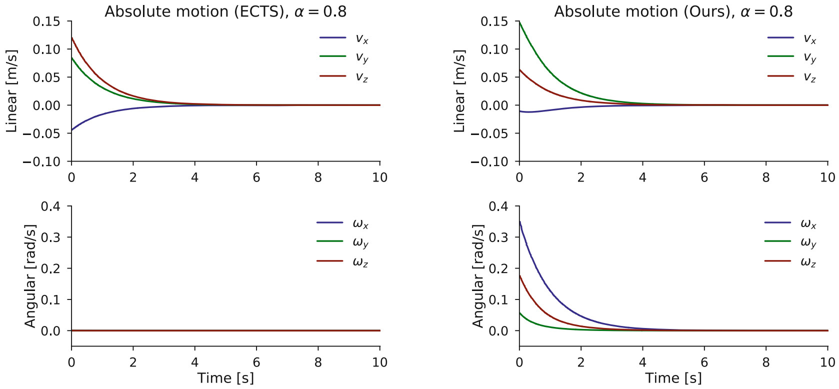
6.1 Nullspace Projection of the Extended Relative Jacobian Solution



(a) ECTS (b) Proposed

Fig. 7. Results for the execution of a pure **rotational** task.

In this case study, we show how the joint space solution (32) introduces undesirable relative motion components, as a consequence of the asymmetric orthogonality from Eq. (19). This property implies that asymmetric absolute motion lies in the nullspace of (31), which contains relative motion components, Eq. (13). As such, the nullspace projection (28) is required to use our proposed extended relative Jacobian. In the numerical simulations, we set $\alpha = 0.8$ to denote a blended



(a) ECTS

(b) Proposed

Fig. 8. Translational relative task.

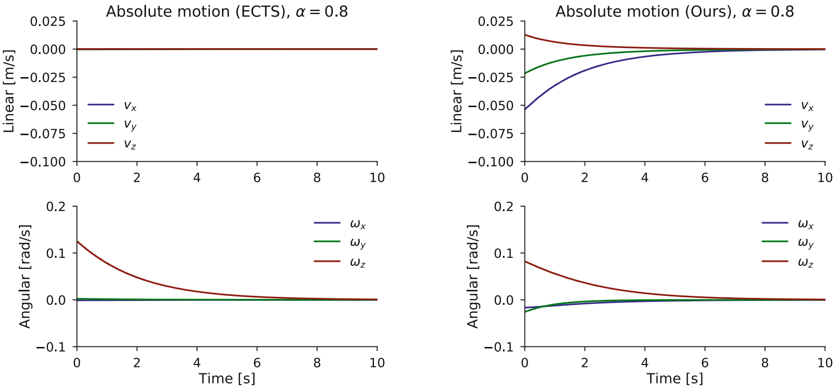
mode where the right manipulator ($i = 2$) executes most of the relative task. Figure 5a depicts the evolution of the relative error for a purely translational task. Conversely, Fig. 5b shows the error progression in a purely rotational task. The solution (32) induces undesired relative motion in both tasks. This is clearly visible in the rotational error for the translational task and in the translational error for the rotational task. The nullspace projection of (32) in (28) filters out these undesired components.

6.2 Asymmetric Relative Motion

The nullspace projection of the asymmetric solution (32) removes some motion components which are redundant to the asymmetric relative task. In this case study, we show that the symmetric absolute motion still remains as an exploitable functional redundancy, and that asymmetric relative motion is achieved, by comparing our method against the ECTS solution with $\mathbf{v}_a = \mathbf{0}$, for different values of the coefficient α. Note that when $\alpha = 0.5$, we obtain the default CTS and relative Jacobian solutions, respectively.

We depict the joint solutions when $\alpha = 0.8$ in Fig. 6, for the translational relative motion task and in Fig. 7 for the rotational task, where it can be observed that the asymmetric task execution results in distinct final configurations for the system. The norm of the joint space trajectory $\int ||\dot{\mathbf{q}}|| \mathrm{d}t$ was 1.48 for the ECTS solution and 0.84 for ours, showing that the availability of the larger redundant space allows for a smaller norm of joint velocities to be computed by the differential IK method. In addition, we show the induced symmetrical absolute motion in both cases in Fig. 8 and 9. We provide the source code required to test other values of α[3].

[3] https://github.com/diogoalmeida/asymmetric_manipulation.



(a) ECTS

(b) Proposed

Fig. 9. Rotational relative task.

The smaller joint trajectory norm of our method is a result of the exploitation of the absolute motion as a functional redundancy. In the translational motion task our method changes the absolute orientation of the system, Fig. 8b, and conversely a linear component is introduced in the rotational motion task, Fig. 9b. In comparison, ECTS will induce symmetrical absolute motion on the system when $\alpha \neq 0.5$, Fig. 8a and 9a, however, this is only along the dimensions commanded by $\mathbf{v}_r$, as noted in Eq. (22).

7 Conclusions

We study how existing methods to model the differential kinematics of dual-armed robotic systems distribute the relative motion between end-effectors, in the context of cooperative manipulation tasks. The asymmetric execution of a relative motion task can be achieved by redefining the absolute motion space, as in the ECTS formulation, or through the addition of a secondary task in a relative Jacobian setting. The ECTS approach requires absolute motion to be part of the system's primary task, and relies on an unintuitive definition of an asymmetric absolute motion space, which is useful solely for the purpose of the asymmetric execution of the relative task. In a relative Jacobian setting, adding a secondary task induces time-varying asymmetries which are a function of the secondary task's definition, as we illustrated in Example 1.

In contrast, we show how a deliberate degree of asymmetry can be imposed on the execution of a relative motion task within a relative Jacobian formulation. This approach allows for, e.g., a master-slave execution of the relative motion task, as opposed to the time-varying asymmetric execution of other relative Jacobian methods. We illustrate the properties of this approach with two case studies. In the first study, we show that the asymmetric absolute motion which lies in the nullspace of our proposed Jacobian includes undesirable relative motion components, and we illustrate how our proposed differential IK method filters out these components. Then, in our second case study, we show how the absolute motion space is retained as a functional redundancy even after applying our differential IK scheme, through a comparison with the ECTS approach: for the same relative motion task, our method obtains smaller joint velocity norms, thanks to the induced absolute motion.

References

1. Ajoudani, A., Tsagarakis, N.G., Lee, J., Gabiccini, M., Bicchi, A.: Natural redundancy resolution in dual-arm manipulation using configuration dependent stiffness (CDS) control. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1480–1486 (2014). https://ieeexplore.ieee.org/document/6907047
2. Almeida, D., Karayiannidis, Y.: Folding assembly by means of dual-arm robotic manipulation. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3987–3993 (2016). https://ieeexplore.ieee.org/document/7487588
3. Almeida, D., Karayiannidis, Y.: Cooperative manipulation and identification of a 2-DOF articulated object by a dual-arm robot. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–5 (2018). https://ieeexplore.ieee.org/document/8460511
4. Almeida, D., Viña, F.E., Karayiannidis, Y.: Bimanual folding assembly: switched control and contact point estimation. In: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 210–216 (2016). https://ieeexplore.ieee.org/document/7803279
5. Caccavale, F., Chiacchio, P., Chiaverini, S.: Task-space regulation of cooperative manipulators. Automatica **36**(6), 879–887 (2000). http://www.sciencedirect.com/science/article/pii/S0005109899002150
6. Ceriani, N.M., Zanchettin, A.M., Rocco, P.: Collision avoidance with task constraints and kinematic limitations for dual arm robots. In: Menegatti, E., Michael, N., Berns, K., Yamaguchi, H. (eds.) Intelligent Autonomous Systems 13. AISC, vol. 302, pp. 1285–1299. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-08338-4_93
7. Chiacchio, P., Chiaverini, S., Siciliano, B.: Cooperative control schemes for multiple robot manipulator systems. In: Proceedings 1992 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2218–2223 (1992). https://ieeexplore.ieee.org/document/219928
8. Chiacchio, P., Chiaverini, S., Siciliano, B.: Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. J. Dyn. Syst. Measur. Control **118**(4), 691–697 (1996). http://dynamicsystems.asmedigitalcollection.asme.org/article.aspx?articleid=1407304
9. Cruciani, S., Smith, C., Kragic, D., Hang, K.: Dexterous manipulation graphs. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2040–2047 (2018). https://ieeexplore.ieee.org/document/8594303
10. Escande, A., Mansard, N., Wieber, P.B.: Hierarchical quadratic programming: fast online humanoid-robot motion generation. Int. J. Robot. Res. **33**(7), 1006–1028 (2014). https://journals.sagepub.com/doi/abs/10.1177/0278364914521306
11. Foresi, G., et al.: An avoidance control strategy for joint-position limits of dual-arm robots. IFAC-PapersOnLine **50**(1), 1056–1061 (2017). 20th IFAC World Congress. http://www.sciencedirect.com/science/article/pii/S2405896317305293

12. Hu, Y., Huang, B., Yang, G.Z.: Task-priority redundancy resolution for cooperative control under task conflicts and joint constraints. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2398–2405 (2015). https://ieeexplore.ieee.org/document/7353702
13. Jamisola, R.S., Roberts, R.G.: A more compact expression of relative Jacobian based on individual manipulator Jacobians. Robot. Auton. Syst. **63**, 158–164 (2015). http://www.sciencedirect.com/science/article/pii/S0921889014001699d
14. Lee, J., Chang, P., Gweon, D.: A cost function inspired by human arms movement for a bimanual robotic machining. In: 2012 IEEE International Conference on Robotics and Automation, pp. 5431–5436 (2012). https://ieeexplore.ieee.org/document/6224643
15. Lee, J., Chang, P.H.: Redundancy resolution for dual-arm robots inspired by human asymmetric bimanual action: formulation and experiments. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6058–6065 (2015). https://ieeexplore.ieee.org/document/7140049
16. Lewis, C.L.: Trajectory generation for two robots cooperating to perform a task. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 2, pp. 1626–1631 (1996). https://ieeexplore.ieee.org/document/506945
17. Lewis, C.L., Maciejewski, A.A.: Trajectory generation for cooperating robots. In: 1990 IEEE International Conference on Systems Engineering, pp. 300–303 (1990). https://ieeexplore.ieee.org/abstract/document/5725688
18. Luh, J., Zheng, Y.: Constrained relations between two coordinated industrial robots for motion control. Int. J. Robot. Res. **6**(3), 60–70 (1987). https://journals.sagepub.com/doi/abs/10.1177/027836498700600304
19. Mohri, A., Yamamoto, M., Hirano, G.: Cooperative path planning for two manipulators. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2853–2858 (1996). https://ieeexplore.ieee.org/document/506595
20. Nemec, B., Likar, N., Gams, A., Ude, A.: Bimanual human robot cooperation with adaptive stiffness control. In: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 607–613 (2016). https://ieeexplore.ieee.org/abstract/document/7803337
21. Ortenzi, D., Muthusamy, R., Freddi, A., Monteriù, A., Kyrki, V.: Dual-arm cooperative manipulation under joint limit constraints. Robot. Auton. Syst. **99**, 110–120 (2018). http://www.sciencedirect.com/science/article/pii/S0921889017300672
22. Owen, W., Croft, E., Benhabib, B.: A multi-arm robotic system for optimal sculpting. Robot. Comput.-Integr. Manuf. **24**(1), 92–104 (2008). http://www.sciencedirect.com/science/article/pii/S0736584506001050
23. Owen, W.S., Croft, E.A., Benhabib, B.: Acceleration and torque redistribution for a dual-manipulator system. IEEE Trans. Robot. **21**(6), 1226–1230 (2005). https://ieeexplore.ieee.org/document/1549951
24. Park, H.A., Lee, C.S.G.: Extended cooperative task space for manipulation tasks of humanoid robots. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6088–6093 (2015). https://ieeexplore.ieee.org/abstract/document/7140053
25. Park, H.A., Lee, C.S.G.: Dual-arm coordinated-motion task specification and performance evaluation. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 929–936 (2016). https://ieeexplore.ieee.org/document/7759161

26. Tarbouriech, S., Navarro, B., Fraisse, P., Crosnier, A., Cherubini, A., Sallé, D.: Dual-arm relative tasks performance using sparse kinematic control. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6003–6009 (2018). https://ieeexplore.ieee.org/abstract/document/8594320
27. Uchiyama, M., Dauchez, P.: A symmetric hybrid position/force control scheme for the coordination of two robots. In: Proceedings 1988 IEEE International Conference on Robotics and Automation, vol. 1, pp. 350–356 (1988). https://ieeexplore.ieee.org/document/12073
28. Uchiyama, M., Iwasawa, N., Hakomori, K.: Hybrid position/force control for coordination of a two-arm robot. In: Proceedings 1987 IEEE International Conference on Robotics and Automation, vol. 4, pp. 1242–1247 (1987). https://ieeexplore.ieee.org/document/1087766
29. Yan, L., Mu, Z., Xu, W., Yang, B.: Coordinated compliance control of dual-arm robot for payload manipulation: master-slave and shared force control. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2697–2702 (2016). https://ieeexplore.ieee.org/document/7759419
30. Zheng, Y.F.: Kinematics and dynamics of two industrial robots in assembly. In: Proceedings 1989 International Conference on Robotics and Automation, vol. 3, pp. 1360–1365 (1989). https://ieeexplore.ieee.org/document/100169