Asymmetric Dual-Arm Task Execution using an Extended Relative Jacobian

Diogo Almeida[1] and Yiannis Karayiannidis[2]

***Abstract*—Dual-armed coordination is a classical problem in robotics, which gained an added relevance as robots are expected to become more autonomous and deployed outside structured environments. Classical solutions to the coordination problem include leader-follower or master-slave approaches, where the coordinated task is assigned to one arm, while the other adopts a passive role, such as the regulation of contact forces. Alternatively, cooperative task space approaches represent the task space of the dual-armed system in terms of absolute and relative motion components, where, for the most part, the relative task is split evenly between the manipulators. Relative Jacobian methods offer an alternative approach where the task space of the system contains only its relative motion. This is an attractive approach for tasks which do not require an absolute motion target, as a larger degree of redundancy becomes available for the optimization of secondary goals of the cooperative system. However, existing relative Jacobian solutions do not allow for a particular degree of collaboration to be set explicitly. In this work, we present a task-space analysis of common cooperation strategies and propose a novel, asymmetric, definition for the relative motion space. We show how this definition enables a user to prescribe a specific degree of cooperation between arms while using a relative Jacobian solution. Simulation results are provided to illustrate some properties of this novel cooperation strategy.**

I. Introduction

Early works on the problem of coordinating two arms addresses it through a leader-follower [19] formulation, or, alternatively, by distributing the task between arms [29]. The latter approach results from the assumption that both arms are jointly grasping an object, which constrains the admissible motion of each manipulator. A mapping of the task to each manipulator needs to ensure that the internal forces exerted on the object remain bounded, and as such a cooperation strategy can be derived from the statics of the system. The principle of virtual work is then applied to derive meaningful kinematics, namely in terms of the *absolute* and *relative* motion of the kinematic chain [29, 28]. The derived inverse kinematics offer a *symmetric* solution to the problem of coordination between the two arms: the obtained absolute and relative motion is evenly shared.

Later work formulated the cooperative kinematics independently of the concrete manipulation task. The resulting strategy is often dubbed Cooperative Task Space (CTS) [9]. Within CTS frameworks, the user specifies a target to the dual-arm system directly in terms of desired absolute and relative motion of the manipulators. In recent years, CTS has been extended to support *asymmetric* motion [25]. This Extended CTS (ECTS) formulation redefines the absolute motion frames of the cooperative task space. The new definition constrains the inverse solution, such that, when resolving the cooperative task into the robotic system's end-effectors, the relative motion task is split asymmetrically between arms. The user can choose the degree of cooperation between the arms by setting a parameter, allowing master-slave, symmetric and asymmetric solutions to the relative task.

Many dual-armed tasks, such as machining or the assembly of two components, can be modelled in terms of the just the relative motion of two manipulators. Therefore, while this sub-set of robotic problems can be solved within the context of CTS-based manipulation frameworks, this over-constrains the problem, as the absolute motion is a functional redundancy to these types of tasks, and CTS requires the specification of absolute motion as part of the primary task definition. Alternatively, we can work exclusively with the relative task space as the primary task space. This results in differential kinematics modelled by a *relative* Jacobian, to which the absolute motion task is indeed a functional redundancy. The larger level of redundancy available to a relative Jacobian solution has been explored to, e.g., avoid self-collisions or workspace obstacles in a task-priority approach [20].

In this article, we perform a task-space analysis of CTS-based methods. We show that the asymmetric formulation in [25] defines motion spaces which are not orthogonal, unlike the original CTS formulation. This observation leads to the definition of an asymmetric relative motion space, which enables the introduction of asymmetry without forcing the user to specify an absolute motion task. We then construct an asymmetric relative Jacobian and show how it generalizes earlier relative Jacobian approaches by enabling a choice of the degree of cooperation between arms on the relative motion task. We illustrate some properties of this approach through a set of case studies.

II. Related Work

A common solution to the problem of cooperation between two robotic arms is to employ leader-follower (or

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

[2]Dept. of Electrical Eng., Chalmers University of Technology, SE-412 96 Gothenburg, Sweden yiannis@chalmers.se

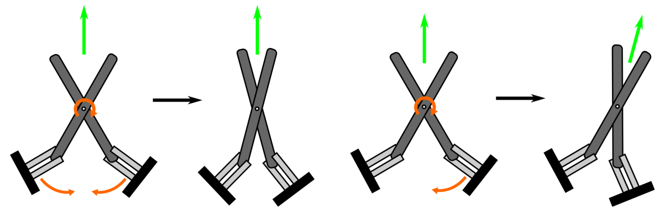
master-slave) approaches [19, 31]. When addressing the coordinated task in a master-slave manner, one arm is usually responsible for the execution of the relative motion required to fullfill the shared task, while the other (the slave, or follower) will implement a passive control policy, typically adopting a compliant behavior. Some recent examples of this completely asymmetric formulation are [30, 3, 5].

Alternatively, the task can be shared between the manipulators. Uchiyama et al. [29] and Uchiyama and Dauchez [28] present an early formulation of the task space division for a dual-armed system in terms of absolute and relative motion components, as a dual to external and internal forces on a jointly held rigid object. Chiacchio et al. [8, 9] provide a definition of absolute and relative motion spaces independently of the statics of the dual-armed system. The resulting relationship between task space variables is the CTS approach to the coordination problem. An alternative definition of CTS using dual-quaternions is given by Adorno et al. [1]. CTS is extended by Park and Lee [25, 26], who redefine the absolute motion space to support asymmetric motion of the two manipulators, and show the advantages of choosing the degree of cooperation between arms to optimize task compatibility indices. Recent methods employ the CTS to describe their tasks. These include human-robot interaction settings, Nemec et al. [21], the cooperative manipulation of a mechanism, Almeida and Karayiannidis [4] or the execution of a bimanual dexterous manipulation task, Cruciani et al. [10].

When only the relative motion task is of relevance, relative Jacobian methods model the kinematics of the manipulation problem without considering an absolute motion task. Lewis and Maciejewski [18] illustrate how a relative Jacobian allows for an assembly task to be defined as the primary task of the joint dual-armed system. This work is expanded by Lewis [17] with alternative formulations for the relative Jacobian, and Mohri et al. [20] use the larger nullspace of the relative Jacobian to enable self-collision and obstacle avoidance. In fact, this larger nullspace results from treating the absolute motion as a functional redundancy to the system.

Recent works employ the relative Jacobian for varied tasks. Non-exhaustively, machining is addressed as a relative motion task by Owen et al. [23, 24]. Jamisola and Roberts [14] derive a relative Jacobian for mobile dual-arm manipulators. Lee et al. [16] and Lee and Chang [15] optimize task-compatibility indices for the two arms of their system independently, while Ajoudani et al. [2] perform a joint optimization of the stiffness profile for tasks that naturally benefits from the adoption of different roles for the two arms, such as a peg-in-hole assembly or drawing. Hu et al. [13], Foresi et al. [12] and Ortenzi et al. [22] exploit the availability of a larger nullspace in a relative task definition to avoid joint limits.

Hierarchical quadratic problems (HQP) [11] are an alternative approach to obtain solutions to the problem of inverse differential kinematics. While in this article we focus on pseudo-inverse solutions, the relative Jacobian has been employed in the context of HQP problems by, e.g., Ceriani et al. [7] and Tarbouriech et al. [27].



(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.

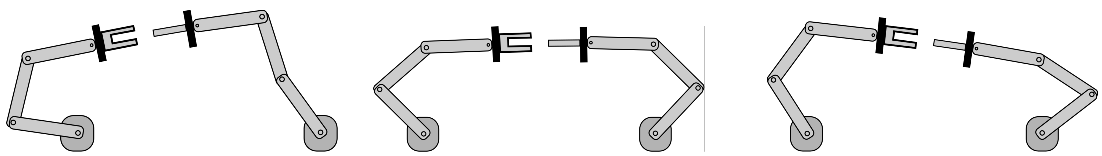
III. Cooperative motion spaces

In this section, we do a task-space analysis of existing cooperation resolution 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 space. The cooperative task is prescribed in these spaces, and can be resolved into velocities for the two end-effectors. A possible approach to obtain *asymmetric* relative motion is to redefine the absolute motion space, as is the case with ECTS. As symmetric relative motion would conflict with the absolute task in the redefined space, the relative solution is constrained to be asymmetric. In contrast, we show how it is possible to achieve the desired asymmetric relative motion through a redefinition of the relative motion space. The resulting space is orthogonal to the ECTS absolute motion space, and allows for an asymmetric relative solution to be obtained without required the specification of an absolute motion to the system.

A. Notation

Consider a dual-armed system composed by two robotic manipulators. Let $\{h_i\}$ denote a generic coordinate frame, e.g., of manipulator's i end-effector, $i = 1, 2$. Each frame is defined by the end-effector's position, $\mathbf{p}_i \in \mathbb{R}^3$ and orientation $\mathbf{R}_i \in \mathcal{SO}(3)$, expressed in a common base frame. The orientation can be represented by an angle and an axis. 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 $\mathcal{N}(\mathbf{A})$.



(a) (b) (c)

Fig. 2: The absolute pose of the cooperative system is redundant w.r.t the relative motion task. In all the three examples, the assembly task can be achieved through the same relative motion of the peg and the hole.

B. Absolute and relative motion spaces

The goal of a cooperative task formulation is to resolve a desired cooperative motion of the system in terms of $\mathbf{v}_1$ and $\mathbf{v}_2$. Hence, to completely specify the cooperative task, 12 degrees-of-freedom are needed. It is convenient to define motion frames which are representative of the cooperative task.

1) Cooperative Task Space: In CTS [9], the user specifies desired *absolute* and *relative* velocities to the system, such that the absolute motion represents the components of $\mathbf{v}_1$ and $\mathbf{v}_2$ along the same direction, and the relative motion is defined as the motion of $\{h_1\}$ with respect to $\{h_2\}$. Absolute and relative motion frames are defined, respectively $\{h_a\}$ and $\{h_r\}$, such that

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

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

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

Absolute and relative motion twists can be obtained through differentiation,

$$\mathbf{v}_a = \frac{1}{2}\begin{bmatrix} \dot{\mathbf{p}}_1 + \dot{\mathbf{p}}_2 \\ \boldsymbol{\omega}_1 + \boldsymbol{\omega}_2 \end{bmatrix} \quad \mathbf{v}_r = \begin{bmatrix} \dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1 \\ \boldsymbol{\omega}_2 - \boldsymbol{\omega}_1 \end{bmatrix}. \tag{3}$$

The user can thus specify a cooperative task in terms of absolute and relative motion. In task space, this relation can be expressed through 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 between them.

2) Extended Cooperative Task Space: The CTS formulation has been extended by Park and Lee [25, 26]. A different definition for the absolute frame is adopted,

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

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),

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

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 either the absolute or the relative motion creates a conflict with the other motion space's symmetric definition, 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 $\mathbf{L}_a, \mathbf{L}_r, \boldsymbol{\mathcal{L}}_a(\alpha) \in \mathbb{R}^{6\times 12}$. As seen previously, 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 definition of absolute motion, $\mathbf{v}_a = \mathbf{L}_a\mathbf{v}$ while ECTS makes use of the asymmetric formulation, $\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\mathbf{v}$.

Remark 1. *The symmetric linking matrices $\mathbf{L}_a$ and $\mathbf{L}_r$ define orthogonal motion spaces. In particular, from* (4) *we can observe that the columns of $\mathbf{L}_{cts}$ are orthogonal or, equivalently,*

$$\mathbf{L}_a\mathbf{L}_r^\dagger = \frac{1}{4}\begin{bmatrix} \mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix} \begin{bmatrix} -\mathbf{I}_6 \\ \mathbf{I}_6 \end{bmatrix} = \mathbf{0}, \tag{10}$$

where the symbol † denotes the Moore-Penrose pseudoinverse of a matrix, i.e.

$$\mathbf{L}_r^\dagger = \mathbf{L}_r^\top(\mathbf{L}_r\mathbf{L}_r^\top)^{-1}. \tag{11}$$

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].

3) 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 (11) to resolve a desired relative motion into velocities of each robot end-effector,

$$\mathbf{v} = \mathbf{L}_r^\dagger \mathbf{v}_r = \frac{1}{2}\begin{bmatrix} -\mathbf{I}_6 \\ \mathbf{I}_6 \end{bmatrix} \mathbf{v}_r, \tag{12}$$

which matches the relative part of the CTS solution (5).

Remark 2. *When commanding symmetric relative motion through* (12)*, the resulting task space velocities* $\mathbf{v}$ *will not contain symmetric absolute motion. This is a consequence of Remark 1. Therefore, the symmetric absolute motion frame is invariant during the execution of the symmetric relative motion task. Equivalently, this means that* $\mathbf{v}_r \in \mathcal{N}(\mathbf{L}_a)$ *and* $\mathbf{v}_a \in \mathcal{N}(\mathbf{L}_r)$*. Thus, symmetric absolute motion can be commanded to the system without disturbing the symmetric relative motion task.*

A secondary task can be added to the solution (12). This will, however, result in a mapping to the symmetric absolute motion space.

Example 1 (Assigning a secondary task)**.** *Let* $\mathbf{v}_{1_d}$ *be a secondary task we wish to assign to end-effector* $i = 1$*. This can be done through a task priority formulation,*

$$\begin{aligned} \mathbf{v} &= \mathbf{L}_r^\dagger \mathbf{v}_r + (\mathbf{I}_{12} - \mathbf{L}_r^\dagger \mathbf{L}_r)[\mathbf{I}_6 \quad \mathbf{0}_6]^\dagger \mathbf{v}_{1_d} \\ &= \frac{1}{2}\begin{bmatrix} \mathbf{v}_{1_d} - \mathbf{v}_r \\ \mathbf{v}_{1_d} + \mathbf{v}_r \end{bmatrix}. \end{aligned} \tag{13}$$

Note that, despite the task being assigned to $\{h_1\}$*,* $\mathbf{v}_{1_d}$ *will be evenly distributed between both end-effectors as a result of the nullspace projection, originating symmetric absolute motion,*

$$\mathbf{v}_a = \mathbf{L}_a \mathbf{v} = \frac{1}{2}\mathbf{v}_{1_d}. \tag{14}$$

IV. Defining an asymmetric relative motion space

In this section, we introduce an asymmetric relative motion frame which define a corresponding asymmetric relative motion space. Together with the differential IK method from section V, this composes the main contribution of our work.

A. Asymmetric relative motion

Consider that we wish to control the relative motion between the end-effectors in an *asymmetric* manner. An extreme example is to define a master-slave relationship between $\{h_1\}$ and $\{h_2\}$ such that only one of the end-effectors is responsible for executing the relative motion task. As shown in Example 1, assigning a secondary task to one of the end-effectors does not enable this scenario and the secondary task is instead mapped to the symmetric absolute motion space.

Analogously to the ECTS approach, we would like instead to be able to specify a degree of cooperation between the arms. This can be achieved by redefining the 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

$$\begin{aligned} \mathbf{p}_r &= \frac{\alpha \mathbf{p}_2 - (1-\alpha)\mathbf{p}_1}{(1-\alpha)^2 + \alpha^2} \\ \mathbf{R}_r &= \mathbf{R}_{\mathbf{k}_1}^\top \left(\frac{(1-\alpha)\vartheta_1}{(1-\alpha)^2 + \alpha^2} \right) \mathbf{R}_{\mathbf{k}_2} \left(\frac{\alpha \vartheta_2}{(1-\alpha)^2 + \alpha^2} \right). \end{aligned} \tag{15}$$

Asymmetric relative motion can be obtained through differentiation of (15),

$$\mathbf{v}_r = \boldsymbol{\mathcal{L}}_r(\alpha)\mathbf{v}, \tag{16}$$

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

$$\boldsymbol{\mathcal{L}}_r(\alpha) = \frac{1}{(1-\alpha)^2 + \alpha^2}\begin{bmatrix} -(1-\alpha)\mathbf{I}_6 & \alpha \mathbf{I}_6 \end{bmatrix}. \tag{17}$$

The particular solution to (16) is analogous to (12) and corresponds to the the ECTS inverse mapping (8) ,

$$\mathbf{v} = \boldsymbol{\mathcal{L}}_r(\alpha)^\dagger \mathbf{v}_r = \begin{bmatrix} -(1-\alpha)\mathbf{I}_6 \\ \alpha \mathbf{I}_6 \end{bmatrix} \mathbf{v}_r. \tag{18}$$

Theorem 1. *The motion space defined by* $\{h_r\}$ *in* (15) *is characterized by the following properties:*

1) *Commanding a relative velocity in* $\{h_r\}$*, eq.* (18)*, 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

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

and consider the solution (18). We have,

$$\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\boldsymbol{\mathcal{L}}_r(\alpha)^\dagger \mathbf{v}_r = \mathbf{0}, \tag{20}$$

and thus, the asymmetric absolute motion frame (6) remains invariant to velocities commanded in the motion frame from (15). $\boldsymbol{\mathcal{L}}_r(\alpha)$ acts as a generalized inverse to $\mathbf{L}_r$ since

$$\mathbf{L}_r \boldsymbol{\mathcal{L}}_r(\alpha)^\dagger = \begin{bmatrix} -\mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix} \begin{bmatrix} -(1-\alpha)\mathbf{I}_6 \\ \alpha \mathbf{I}_6 \end{bmatrix} = \mathbf{I}_6. \tag{21}$$

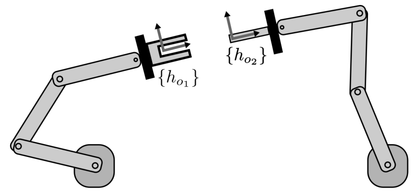


Fig. 3: An assembly task can be represented as the relative motion of two frames, rigidly attached to each arm's end-effector.

□

A consequence of (21) is that the cooperative solution (18) can constrain (12) when added in a task-priority scheme, i.e.,

$$\mathbf{v} = \mathbf{L}_r^\dagger \mathbf{v}_r + (\mathbf{I}_{12} - \mathbf{L}_r^\dagger \mathbf{L}_r)\boldsymbol{\mathcal{L}}_r(\alpha)^\dagger \mathbf{v}_r = \boldsymbol{\mathcal{L}}_r(\alpha)^\dagger \mathbf{v}_r. \quad (22)$$

Unlike the approach in Example 1, this enables a specific degree of cooperation to be set by the user. In particular, $\alpha = 0$ or $\alpha = 1$ results in a master-slave task distribution, with respectively $\{h_1\}$ and $\{h_2\}$ executing the entirety of the relative motion task.

Remark 3. *Consider a commanded absolute or relative motion in the redefined motion spaces* (6) *and* (15). *The end-effectors' velocities obtained through the inverse mappings* $\boldsymbol{\mathcal{L}}_a(\alpha)^\dagger$ *and* $\boldsymbol{\mathcal{L}}_r(\alpha)^\dagger$ *will contain symmetric relative and absolute motion components, respectively,*

$$\mathbf{v}_r = \mathbf{L}_r \boldsymbol{\mathcal{L}}_a(\alpha)^\dagger \mathbf{v}_a = \frac{1-2\alpha}{\alpha^2 + (1-\alpha)^2}\mathbf{v}_a \quad (23)$$

$$\mathbf{v}_a = \mathbf{L}_a \boldsymbol{\mathcal{L}}_r(\alpha)^\dagger \mathbf{v}_r = \frac{2\alpha - 1}{2}\mathbf{v}_r. \quad (24)$$

The converse is true as well,

$$\mathbf{v}_a = \boldsymbol{\mathcal{L}}_a(\alpha)\mathbf{L}_r^\dagger \mathbf{v}_r = \frac{1-2\alpha}{2}\mathbf{v}_r \quad (25)$$

$$\mathbf{v}_r = \boldsymbol{\mathcal{L}}_r(\alpha)\mathbf{L}_a^\dagger \mathbf{v}_a = \frac{2\alpha - 1}{\alpha^2 + (1-\alpha)^2}\mathbf{v}_a. \quad (26)$$

V. 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 Inverse Kinematics (IK) separately. Alternatively, we can derive IK algorithms which directly resolve a desired cooperative motion to the joint state of the complete dual-arm chain.

A. Notation

Let $\mathbf{q}_i \in \mathbb{R}^n$ represent the joint variables of the i-th manipulator, with $n \geq 6$. Its differential kinematics are obtained through the relation $\mathbf{v}_i = \mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i$, where $\mathbf{J}_i(\mathbf{q}_i) \in \mathbb{R}^{6\times n}$ is the manipulator's Jacobian. 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 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

B. 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 section III and computing the differential IK for each individual manipulator. As an example, a relative motion task can be solved into joint space by computing

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \mathbf{L}_r^\dagger \mathbf{v}_r, \quad (27)$$

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 cooperative motion definitions previously discussed 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, $\{h_{o_i}\}$, Fig. 3. 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,

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\mathbf{r}_i) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix}, \quad (28)$$

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

$$\mathbf{J}_{cts} = \mathbf{L}_{cts}\mathbf{W}\mathbf{J} \qquad \mathbf{J}_E(\alpha) = \mathbf{L}_E(\alpha)\mathbf{W}\mathbf{J}, \quad (29)$$

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

$$\mathbf{J}_r = \mathbf{L}_r \mathbf{W}\mathbf{J}, \quad (30)$$

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 $\text{rank}(\mathbf{J}_r) \leq 6$ and thus

$$\dim(\mathcal{N}(\mathbf{J}_r)) \geq \dim(\mathcal{N}(\mathbf{J})) = \dim(\mathcal{N}(\mathbf{J}_1)) + \dim(\mathcal{N}(\mathbf{J}_2)). \quad (31)$$

In practice, the larger nullspace includes absolute motion components which are not present when solving (27). The Jacobian in (30) is often called relative Jacobian.

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

$$\mathbf{v}_r = \mathbf{J}_r \dot{\mathbf{q}}. \quad (32)$$

In the following, we will focus on finding feasible joint space solutions for (32). It is well known that such solutions have the general form

$$\dot{\mathbf{q}} = \mathbf{J}_r^\dagger \mathbf{v}_r + (\mathbf{I}_{2n} - \mathbf{J}_r^\dagger \mathbf{J}_r)\boldsymbol{\zeta}, \tag{33}$$

with $(\mathbf{I}_{2n} - \mathbf{J}_r^\dagger \mathbf{J}_r)\boldsymbol{\zeta}$ being part of the homogeneous solution to (32). 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 (30), as seen in (10). This absolute motion is then a functional redundancy to (32) 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, CTS-based solutions (29) are ideal. Alternatively, using (27) 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 [20].

C. Asymmetric relative Jacobian

A common choice for $\boldsymbol{\zeta}$ is to specify a motion for one of the system's end-effectors [2, 14, 13, 12], e.g.,

$$\boldsymbol{\zeta} = [\mathbf{J}_1 \ \mathbf{0}]^\dagger \mathbf{v}_1. \tag{34}$$

As seen in Example 1, this secondary task will be mapped into the system's absolute motion space. If $\mathbf{v}_1$ is, e.g., the result of a feedback controller responsible for assigning a desired pose to $\{h_1\}$, the solution to (33) with (34) as the secondary task will induce asymmetries in the motion of the arms.

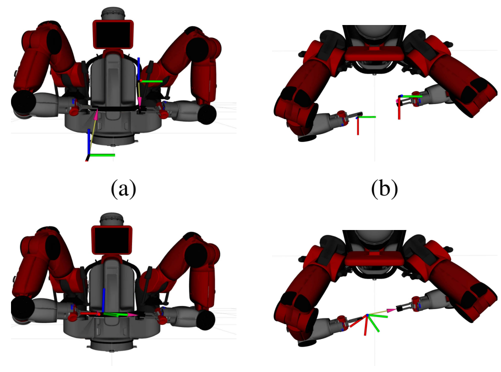
We propose instead to use the asymmetric relative motion space defined in section IV to extend the relative Jacobian methods and allow for setting the degree of cooperation through a parameter, similar to how ECTS extends the CTS formulation. We use (17) to define an asymmetric Jacobian,

$$\mathbf{J}_r(\alpha) = \boldsymbol{\mathcal{L}}_r(\alpha)\mathbf{W}\mathbf{J}. \tag{35}$$

We can now solve the differential IK by defining

$$\boldsymbol{\zeta} = \mathbf{J}_r(\alpha)^\dagger \mathbf{v}_r. \tag{36}$$

Note that imposing the solution in (36) as a secondary task will bear no effect on the desired relative motion, as we have shown in eq. (22). Instead, as we will illustrate in section VI, it will prevent the IK solution to contain undesired asymmetric absolute motion, which can be present in the minimum norm solution (36), due to the asymmetric invariance described in Theorem 1, eq. (20).



(a) (b) (c) (d)

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

D. Relative task

The choice of $\mathbf{v}_r$ in (33) defines the relative motion task. This can be set as the output of a task-specific controller [4], a feedforward command from a teleoperator, or as an error signal from a pose regulation task [6, 26]. 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 [6],

$$\mathbf{v}_r = -\mathbf{K}_p \begin{bmatrix} \tilde{\mathbf{p}} \\ \mathbf{R}_{o_1}\tilde{\boldsymbol{\xi}} \end{bmatrix}, \tag{37}$$

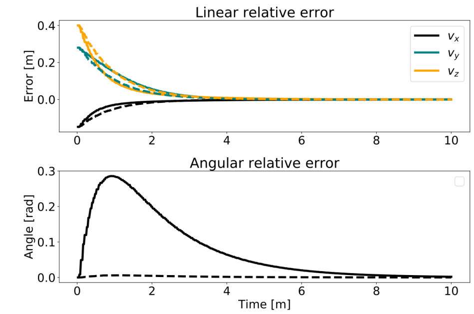
where $\mathbf{K}_p \in \mathbb{R}^{6\times 6}$ is positive definite.

VI. Case Studies

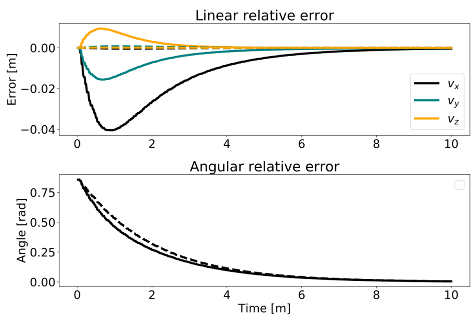
We illustrate different properties of our method with two case studies, in which the primary task is a relative motion task given by (37) with $\mathbf{K}_p = \mathbf{I}_6$. The case studies are based on a simulated Rethink Robotics' Baxter dual-armed robot. Two relative motion tasks are considered, one where the alignment error is purely translational, Fig. 4a-4b, the other where the error is purely rotational, Fig. 4c-4d. The initial object frames poses in the reference frame (Baxter's torso frame) are given in Table I, where the left manipulator is assigned the index $i = 1$ and $i = 2$ denotes the right manipulator. We take the roll, pitch and yaw (RPY) representation of $\mathbf{R}_{o_1}$ and $\mathbf{R}_{o_2}$ for the values in the table.

A. Nullspace projection of the extended relative Jacobian solution

In Theorem 1, we have shown that $\boldsymbol{\mathcal{L}}_r(\alpha)^\dagger$ is a generalized inverse of $\mathbf{L}_r$. The task space solution (18) will thus result in the same relative motion as (12). However, when constructing the extended relative Jacobian in (36),

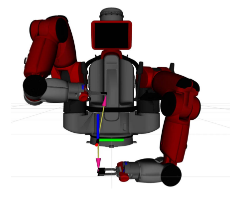


(a) **Translational** relative motion task.

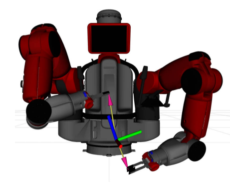


(b) **Rotational** relative motion task.

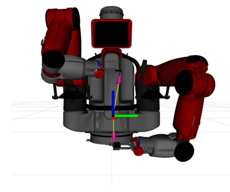
Fig. 5: Evolution of the relative errors in the two types of motion task, when $\alpha = 0.8$. The full lines represent the solution when using $\boldsymbol{\zeta}$ from (36) as a primary task. The dashed lines use (36) in a task priority manner (33).



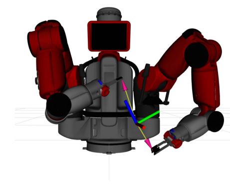
(a) $\alpha = 0.2$



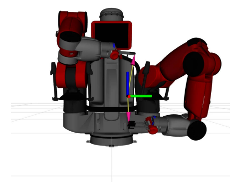
(d) $\alpha = 0.2$



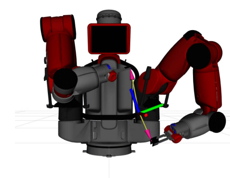
(b) $\alpha = 0.5$



(e) $\alpha = 0.5$



(c) $\alpha = 0.8$



(f) $\alpha = 0.8$

Fig. 6: Pure **translational** relative motion is commanded to the system, which is initialized in the configuration from Fig. 4a. Left column: ECTS solution; right column: our method.

Task type	Initial left object pose	Initial right object pose
Translational	$\mathbf{p} = [0.36, 0.15, 0.36]^\top$ RPY $= [0, 0, 0]^\top$	$\mathbf{p} = [0.51, -0.13, -0.04]^\top$ RPY $= [0, 0, 0]^\top$
Rotational	$\mathbf{p} = [0.45, 0.0, 0.21]^\top$ RPY $= [0.0, 0.0, 0]^\top$	$\mathbf{p} = [0.45, 0.0, 0.21]^\top$ RPY $= [0, 0, -0.8]^\top$

TABLE I: Initial object frames for the case studies.

we introduce a mapping of the commanded motion to the manipulators' joint space and the solution (36) will minimize the norm of the joint space variables. This means that motion in the nullspace of $\boldsymbol{\mathcal{L}}_r(\alpha)$ can be induced, depending on the manipulators' geometry. When $\alpha = 0.5$, the two linking matrices coincide and thus the induced absolute motion is symmetric, as seen in (10). However, in blended mode ($\alpha \neq 0.5$), asymmetric absolute motion can arise, eq. (20). This is undesirable for any relative motion where the task requires contact between two parts, as unregulated asymmetric absolute motion might violate task constraints. As an example, when jointly manipulating a mechanism, we want the commanded relative motion to apply only along the prescribed directions [4].

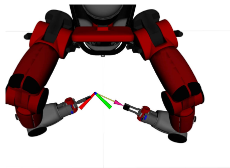
For this case study, we set $\alpha = 0.8$ to denote a blended mode where the right manipulator ($i = 2$) executes most of the relative task. Fig. 5a depicts the evolution of the relative error for a purely translational task. Conversely, Fig. 5b shows the error progression in a purely rotational task. The solution (36) induces undesired relative motion in both tasks. This is clearly visible in the rotation error for the translational task and in the translational error for the rotational task. The nullspace projection of (36) in (33) filters out these undesired components.

B. Asymmetric relative motion

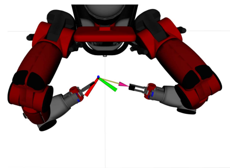
The nullspace projection of the asymmetric solution (36) removes some motion components which are redundant to the asymmetric relative task. In this case study, we show that the symmetric absolute motion still remains as an exploitable functional redundancy, and that asymmetric relative motion is achieved, 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. In our tests, we use the initial object poses from table I, and compute the IK solutions for $\alpha \in \{0.2, 0.5, 0.8\}$.

	Translational task			Rotational task		
α	0.2	0.5	0.8	0.2	0.5	0.8
ECTS	1.49	1.25	1.48	1.46	1.08	1.19
Ours	0.84	0.65	0.84	0.67	0.47	0.56

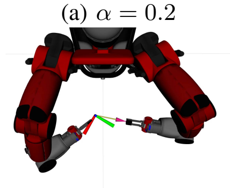
TABLE II: Norm of the joint space velocities over the duration of the relative motion tasks, for different values of α.



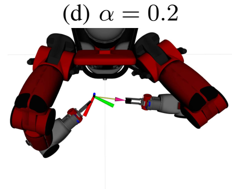
(a) $\alpha = 0.2$



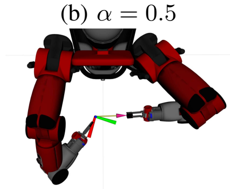
(d) $\alpha = 0.2$



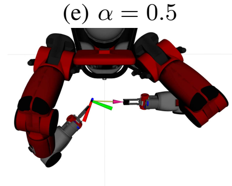
(b) $\alpha = 0.5$



(e) $\alpha = 0.5$



(c) $\alpha = 0.8$



(f) $\alpha = 0.8$

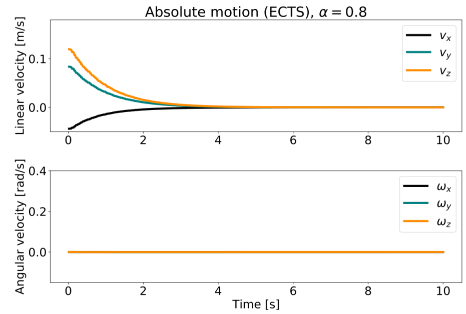
Fig. 7: Pure **rotational** relative motion is commanded to the system, which is initialized in the configuration from Fig. 4d. Left column: ECTS solution; right column: our method.

Solutions for both methods are depicted 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 $\int ||\dot{\mathbf{q}}||\mathrm{d}t$ of both methods is reported on table II. In addition, we show the induced symmetrical absolute motion in both cases, when $\alpha = 0.8$, in Fig. 8 and 9.

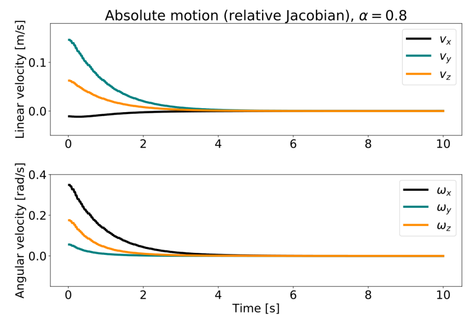
The smaller joint trajectory norm of our method is a result of the exploitation of the absolute motion as a functional redundancy. In the translational motion task our method changes the absolute orientation of the system, Fig. 8b, and conversely a linear component is introduced in the rotational motion 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 according to eq. (24).

VII. CONCLUSIONS

In this article, we introduced an extended relative Jacobian which enables a user to set a specific degree of cooperation in a relative motion task executed by a dual-armed robotic system. The new Jacobian results from an asymmetric relative motion space definition. We have shown that this asymmetric relative space is orthogonal to the ECTS absolute motion space and generalizes the symmetric relative motion space traditionally used in a relative Jacobian setting. The relative Jacobian formulation allows a user to exploit the absolute motion of the system as a functional redundancy, in contrast with CTS based methods, where the

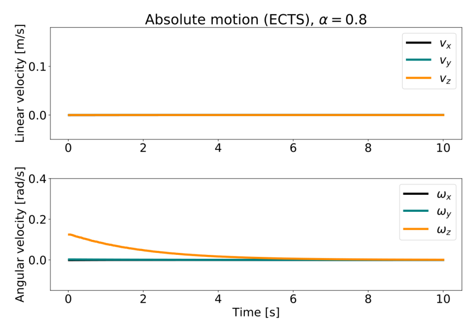


(a) ECTS

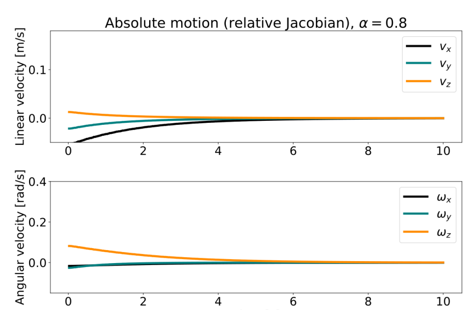


(b) Our method.

Fig. 8: Symmetrical absolute motion induced by the ECTS solution and by our approach during execution of the **translational** relative task.



(a) ECTS



(b) Our method.

Fig. 9: Symmetrical absolute motion induced by the ECTS solution and by our approach during execution of the **rotational** relative task.

absolute motion is part of the primary task definition. With our method, we are able to induce asymmetric task execution while keeping the absolute motion functionally redundant. We present a set of case studies which showcase properties of this method. The method is applicable to any task that can be solved with a relative Jacobian approach, and we will propose novel algorithms to set the degree of cooperation in future work.

References

[1] B. V. Adorno, P. Fraisse, and S. Druon. Dual position control strategies using the cooperative dual task-space framework. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3955–3960, Oct 2010. doi: 10.1109/IROS.2010.5650218. URL https://ieeexplore.ieee.org/document/5650218.

[2] A. Ajoudani, N. G. Tsagarakis, J. Lee, M. Gabiccini, and A. Bicchi. Natural redundancy resolution in dual-arm manipulation using configuration dependent stiffness (CDS) control. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1480–1486, May 2014. doi: 10.1109/ICRA.2014.6907047. URL https://ieeexplore.ieee.org/document/6907047.

[3] D. Almeida and Y. Karayiannidis. Folding assembly by means of dual-arm robotic manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3987–3993, May 2016. doi: 10.1109/ICRA.2016.7487588. URL https://ieeexplore.ieee.org/document/7487588.

[4] D. Almeida and Y. Karayiannidis. 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)*, pages 1–5, May 2018. doi: 10.1109/ICRA.2018.8460511. URL https://ieeexplore.ieee.org/document/8460511.

[5] D. Almeida, F. E. Viña, and Y. Karayiannidis. Bimanual folding assembly: Switched control and contact point estimation. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 210–216, Nov 2016. doi: 10.1109/HUMANOIDS.2016.7803279. URL https://ieeexplore.ieee.org/document/7803279.

[6] F. Caccavale, P. Chiacchio, and S. Chiaverini. Task-space regulation of cooperative manipulators. *Automatica*, 36(6):879 – 887, 2000. ISSN 0005-1098. doi: https://doi.org/10.1016/S0005-1098(99)00215-0. URL http://www.sciencedirect.com/science/article/pii/S0005109899002150.

[7] Nicola Maria Ceriani, Andrea Maria Zanchettin, and Paolo Rocco. Collision Avoidance with Task Constraints and Kinematic Limitations for Dual Arm Robots. In Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi, editors, *Intelligent Autonomous Systems 13*, pages 1285–1299, Cham, 2016. Springer International Publishing. ISBN 978-3-319-08338-4. URL https://link.springer.com/chapter/10.1007/978-3-319-08338-4_93.

[8] P. Chiacchio, S. Chiaverini, and B. Siciliano. Cooperative control schemes for multiple robot manipulator systems. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2218–2223 vol.3, May 1992. doi: 10.1109/ROBOT.1992.219928. URL https://ieeexplore.ieee.org/document/219928.

[9] P Chiacchio, S Chiaverini, and Bruno Siciliano. Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *Journal of dynamic systems, measurement, and control*, 118(4):691–697, 1996. URL http://dynamicsystems.asmedigitalcollection.asme.org/article.aspx?articleid=1407304.

[10] S. Cruciani, C. Smith, D. Kragic, and K. Hang. Dexterous Manipulation Graphs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2040–2047, Oct 2018. doi: 10.1109/IROS.2018.8594303. URL https://ieeexplore.ieee.org/document/8594303.

[11] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014. doi: 10.1177/0278364914521306. URL https://journals.sagepub.com/doi/abs/10.1177/0278364914521306.

[12] G. Foresi, A. Freddi, V. Kyrki, A. Monteriù, R. Muthusamy, D. Ortenzi, and D. Proietti Pagnotta. An Avoidance Control Strategy for Joint-Position Limits of Dual-Arm Robots. *IFAC-PapersOnLine*, 50(1):1056 – 1061, 2017. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2017.08.217. URL http://www.sciencedirect.com/science/article/pii/S2405896317305293. 20th IFAC World Congress.

[13] Yang Hu, Bidan Huang, and Guang-Zhong Yang. Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2398–2405, Sept 2015. doi: 10.1109/IROS.2015.7353702. URL https://ieeexplore.ieee.org/document/7353702.

[14] Rodrigo S. Jamisola and Rodney G. Roberts. A more compact expression of relative Jacobian based on individual manipulator Jacobians. *Robotics and Autonomous Systems*, 63:158 – 164, 2015. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2014.08.011. URL http://www.sciencedirect.com/science/article/pii/S0921889014001699.

[15] J. Lee and P. H. Chang. 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), pages 6058–6065, May 2015. doi: 10.1109/ICRA.2015.7140049. URL https://ieeexplore.ieee.org/document/7140049.

[16] J. Lee, P. Chang, and D. Gweon. A cost function inspired by human arms movement for a bimanual robotic machining. In *2012 IEEE International Conference on Robotics and Automation*, pages 5431–5436, May 2012. doi: 10.1109/ICRA.2012.6224643. URL https://ieeexplore.ieee.org/document/6224643.

[17] C. L. Lewis. Trajectory generation for two robots cooperating to perform a task. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1626–1631 vol.2, April 1996. doi: 10.1109/ROBOT.1996.506945. URL https://ieeexplore.ieee.org/document/506945.

[18] C. L. Lewis and A. A. Maciejewski. Trajectory generation for cooperating robots. In *1990 IEEE International Conference on Systems Engineering*, pages 300–303, Aug 1990. doi: 10.1109/ICSYSE.1990.203156. URL https://ieeexplore.ieee.org/abstract/document/5725688.

[19] J.Y.S. Luh and Y.F. Zheng. Constrained Relations between Two Coordinated Industrial Robots for Motion Control. *The International Journal of Robotics Research*, 6(3):60–70, 1987. doi: 10.1177/027836498700600304. URL https://journals.sagepub.com/doi/abs/10.1177/027836498700600304.

[20] A. Mohri, M. Yamamoto, and G. Hirano. Cooperative path planning for two manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2853–2858 vol.3, April 1996. doi: 10.1109/ROBOT.1996.506595. URL https://ieeexplore.ieee.org/document/506595.

[21] B. Nemec, N. Likar, A. Gams, and A. Ude. Bimanual human robot cooperation with adaptive stiffness control. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 607–613, Nov 2016. doi: 10.1109/HUMANOIDS.2016.7803337. URL https://ieeexplore.ieee.org/abstract/document/7803337.

[22] Davide Ortenzi, Rajkumar Muthusamy, Alessandro Freddi, Andrea Monteriù, and Ville Kyrki. Dual-arm cooperative manipulation under joint limit constraints. *Robotics and Autonomous Systems*, 99:110 – 120, 2018. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2017.10.005. URL http://www.sciencedirect.com/science/article/pii/S0921889017300672.

[23] W. S. Owen, E. A. Croft, and B. Benhabib. Acceleration and torque redistribution for a dual-manipulator system. *IEEE Transactions on Robotics*, 21(6):1226–1230, Dec 2005. ISSN 1552-3098. doi: 10.1109/TRO.2005.853492. URL https://ieeexplore.ieee.org/document/1549951.

[24] W.S. Owen, E.A. Croft, and B. Benhabib. A multi-arm robotic system for optimal sculpting. *Robotics and Computer-Integrated Manufacturing*, 24(1):92 – 104, 2008. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2006.08.001. URL http://www.sciencedirect.com/science/article/pii/S0736584506001050.

[25] H. A. Park and C. S. G. Lee. Extended Cooperative Task Space for manipulation tasks of humanoid robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6088–6093, May 2015. doi: 10.1109/ICRA.2015.7140053. URL https://ieeexplore.ieee.org/abstract/document/7140053.

[26] H. A. Park and C. S. G. Lee. Dual-arm coordinated-motion task specification and performance evaluation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 929–936, Oct 2016. doi: 10.1109/IROS.2016.7759161. URL https://ieeexplore.ieee.org/document/7759161.

[27] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé. Dual-Arm Relative Tasks Performance Using Sparse Kinematic Control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6003–6009, Oct 2018. doi: 10.1109/IROS.2018.8594320. URL https://ieeexplore.ieee.org/abstract/document/8594320.

[28] M. Uchiyama and P. Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 350–356 vol.1, April 1988. doi: 10.1109/ROBOT.1988.12073. URL https://ieeexplore.ieee.org/document/12073.

[29] M. Uchiyama, N. Iwasawa, and K. Hakomori. Hybrid position/Force control for coordination of a two-arm robot. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1242–1247, March 1987. doi: 10.1109/ROBOT.1987.1087766. URL https://ieeexplore.ieee.org/document/1087766.

[30] L. Yan, Z. Mu, W. Xu, and B. Yang. 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)*, pages 2697–2702, Oct 2016. doi: 10.1109/IROS.2016.7759419. URL https://ieeexplore.ieee.org/document/7759419.

[31] Y. F. Zheng. Kinematics and dynamics of two industrial robots in assembly. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 1360–1365 vol.3, May 1989. doi: 10.1109/ROBOT.1989.100169. URL https://ieeexplore.ieee.org/document/100169.