



HAL
open science

Dual-arm relative tasks performance using sparse kinematic control

Sonny Tarbouriech, Benjamin Navarro, Philippe Fraitse, André Crosnier, Andrea Cherubini, Damien Sallé

► **To cite this version:**

Sonny Tarbouriech, Benjamin Navarro, Philippe Fraitse, André Crosnier, Andrea Cherubini, et al.. Dual-arm relative tasks performance using sparse kinematic control. IROS: Intelligent Robots and Systems, Oct 2018, Madrid, Spain. pp.6003-6008, 10.1109/IROS.2018.8594320 . hal-01735462v2

HAL Id: hal-01735462

<https://hal.science/hal-01735462v2>

Submitted on 14 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dual-arm relative tasks performance using sparse kinematic control

Sonny Tarbouriech*[†] Benjamin Navarro* Philippe Fraisse*
André Crosnier* Andrea Cherubini* and Damien Sallé[†]

Abstract—To make production lines more flexible, dual-arm robots are good candidates to be deployed in autonomous assembly units. In this paper, we propose a sparse kinematic control strategy, that minimizes the number of joints actuated for a coordinated task between two arms. The control strategy is based on a hierarchical sparse QP architecture. We present experimental results that highlight the capability of this architecture to produce sparser motions (for an assembly task) than those obtained with standard controllers.

I. INTRODUCTION

The H2020 VERSATILE European Project¹ aims at contributing to the development of the “Factory of the Future”. In particular, flexibility and versatility are enhanced in production lines, by integrating dual-arm robotic platforms.

Indeed, using multiple cooperative robotic arms significantly increases the potential of manipulators [1]. This type of structures allows the manipulation of large and heavy objects [2] as well as the achievement of difficult tasks that are not feasible with a single arm [3], [4].

However, the resulting system becomes more complex from a control perspective. Arms synchronization has to be perfectly managed throughout a cooperative task, while intrinsic constraints arising from the dual-arm structure reduce the solution space, compromising proper task tracking.

Despite the high number of degrees of freedom (DOF), a dual-arm system is generally “not very redundant”. In fact, the task is often specified by 12 parameters (6 for the pose of each end-effector). Manipulators commonly used in the industry have 6 or 7 DOF, meaning that the overall dual-arm platform will have 12 or 14 DOF. In such cases, the robot is not (or slightly) redundant with respect to the task as all (or almost all) the joints are required to achieve it.

In [5], Chiacchio et al. presented a kinematic representation for dual-arm systems performing coordinated tasks. Their idea was to consider the system as a single entity and to describe the task in terms of cooperative variables, defined by an absolute frame (computed as a function of the two end-effector poses, expressed in the world frame) and a relative frame (expression of one end-effector’s pose with respect to the other).

Many industrial bimanual applications require the coordination of the arms’ relative motions without having to specify the task in the workspace. Notably, this happens when an object is held by one manipulator while the other realizes a

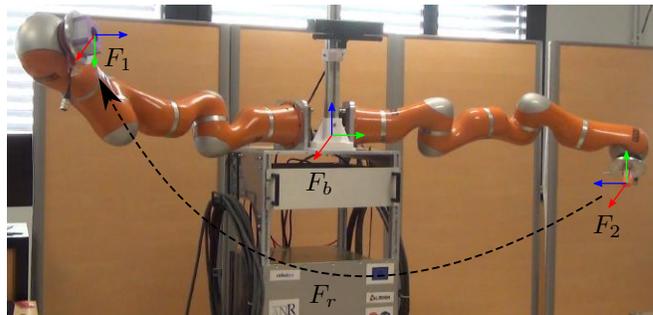


Fig. 1. Representation of the frames related to our dual-arm robot.

manufacturing operation on it with a tool. Examples of such tasks are assembling [6], sculpting [7] or welding [8].

In this case, the cooperative task space representation cited above allows to control the relative frame only, so the variables related to the absolute pose can be removed from the task specification, adding 6 DOF to the task’s null space.

Exploiting redundancy for dual-arm systems has been the topic of several studies, in which available DOF were used for obstacle avoidance [9], to increase manipulability [10] or to satisfy joint constraints [11], [12]. These approaches make use of redundancy by defining a hierarchic prioritized task architecture based on the projection on the null space of the Jacobian [13]. This method guarantees that the tasks of lower priority do not affect the performance of the highest priority task. However, there is *no explicit management of hard constraints*, i.e. constraints that must be satisfied all the time.

A second issue with most approaches is that they impose the *motion of all actuators*. In fact, inverse kinematics is usually solved using the Jacobian pseudo-inverse; yet, on redundant robots, this solution activates more actuators than needed by the task, leading to non-economic movements. This is an undesired behavior in industrial environments, as it increases the potential risk of collision with unexpected obstacles. Moreover, this generally results in a motion that is confusing for an operator working near the robot.

Instead, an approach originally introduced in the signal processing field [14] aims at generating a sparse, or parsimonious, solution. In the context of robotics, a sparse solution is one that activates as few actuators as possible, reducing the overall motion and, consequently, the risk of unintended contact with the environment. It has also been shown that a parsimonious motion provides more natural human-like movements and is therefore more predictable

* LIRMM, Université de Montpellier, CNRS, Montpellier, France. firstname.lastname@lirmm.fr

[†] Tecnalia Research and Innovation, Industry and Transport Division.

¹<https://versatile-project.eu/>

from the operator point of view [15], [16].

In this paper, we propose a parsimonious kinematic control strategy for dual-arm manipulators performing relative tasks under joint limit constraints. Our approach is based on a hierarchy of tasks, implemented as a sequence of quadratic programs (QP) [17]. The main contributions of our framework are:

- The primary task (relative end-effector pose control) is formulated as an optimization problem which returns a sparse solution that minimizes the generated motion, in the actuators' space.
- A secondary task with lower priority is occasionally activated to move joints away from their physical limits.
- Strict task priority is handled: lower-priority tasks are performed as much as possible without producing any effect on higher-priority tasks. Joint position and velocity limits are integrated as hard constraints to strictly respect the robot's physical capabilities.

The paper is organized as follows. Section II formulates the kinematic control scheme for dual-arm coordinated motions. Section III introduces the parsimonious task-solving approach as an optimization problem. Our strategy to take into account joint limits is presented in Sect. IV. A description of the developed QP-based hierarchical architecture is given in Sect. V. Experimental results of a screwing scenario performed on a real dual-arm platform are reported in VI. Finally, Sect. VII ends the paper with concluding remarks and future perspectives.

II. DUAL-ARM KINEMATIC CONTROL LOOP

A. Task representation for coordinated motion

The dual-arm robot that we consider here is depicted in Fig. 1. For each manipulator ($i = 1, 2$), \mathbf{p}_i^b denotes the end-effector position and \mathbf{R}_i^b the rotation matrix, both expressed in a common base frame (superscript "b").

The relative pose corresponds to the coordinates of one end-effector's frame with respect to the other. If we consider F_1 as the reference frame, the relative position is given by:

$$\mathbf{p}_r^1 = \mathbf{p}_2^1. \quad (1)$$

Similarly, the relative orientation can be defined with reference to frame F_1 in terms of the rotation matrix

$$\mathbf{R}_r^1 = \mathbf{R}_2^1. \quad (2)$$

B. The relative Jacobian definition

The standard inverse kinematic problem consists in finding joint velocities $\dot{\mathbf{q}}$ to obtain a desired task space velocity $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}. \quad (3)$$

For a two-manipulator system with relative control, the velocity vectors in the joint and task spaces are defined as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v}_r^1 \\ \boldsymbol{\omega}_r^1 \end{bmatrix} \text{ and } \dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix}, \quad (4)$$

with \mathbf{v}_r^1 , $\boldsymbol{\omega}_r^1$ respectively the linear and angular velocity of the task variables.

Solving the inverse kinematic problem using this formalism induces the definition of a relative Jacobian \mathbf{J} that maps the joint velocities of the two robots to their relative motion in task-space. A first expression was proposed in [18]. Following formulations were derived from the Jacobians of each manipulator (\mathbf{J}_1^b and \mathbf{J}_2^b), leading to simpler and more efficient implementations [19], [20]. However, these expressions omit that the rotational velocity of the reference frame attached to one of the end-effectors can be non-null [21]. Recent work [22] addressed this issue and proposed:

$$\mathbf{J} = [-\boldsymbol{\Psi}_r^1 \boldsymbol{\Omega}_b^1 \mathbf{J}_1^b \quad \boldsymbol{\Omega}_b^1 \mathbf{J}_2^b], \quad (5)$$

with

$$\boldsymbol{\Psi}_r^1 = \begin{bmatrix} \mathbf{I} & -\mathbf{S}(\mathbf{p}_r^1) \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \boldsymbol{\Omega}_b^1 = \begin{bmatrix} \mathbf{R}_b^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b^1 \end{bmatrix}. \quad (6)$$

C. Closed-loop inverse kinematic problem

For closed-loop control, the inverse kinematic problem in (3) is reformulated as a tracking problem, whose input is the desired relative task velocity $\dot{\mathbf{x}}^*$ and whose outputs are the corresponding joint velocities $\dot{\mathbf{q}}$.

Let \mathbf{e} denote the six-dimensional error between the desired and current task velocities, such as:

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}^* - \dot{\mathbf{x}}. \quad (7)$$

This vector can be partitioned into a position (subscript "p") and an orientation (subscript "o") part:

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_o \end{bmatrix}. \quad (8)$$

The definition of relative position error is straightforward:

$$\mathbf{e}_p = \mathbf{p}_r^{1*} - \mathbf{p}_r^1. \quad (9)$$

The definition of an orientation error \mathbf{e}_o was given by Luh et al. [23] and is expressed here as:

$$\mathbf{e}_o = \frac{1}{2} (\mathbf{S}(\mathbf{n}_r^1) \mathbf{n}_r^{1*} + \mathbf{S}(\mathbf{s}_r^1) \mathbf{s}_r^{1*} + \mathbf{S}(\mathbf{a}_r^1) \mathbf{a}_r^{1*}). \quad (10)$$

where $(\mathbf{n}, \mathbf{s}, \mathbf{a})$ and $(\mathbf{n}^*, \mathbf{s}^*, \mathbf{a}^*)$ denote respectively the actual and desired unit vector triples of the task frame.

Using (3) yields :

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}^* - \mathbf{J}\dot{\mathbf{q}}, \quad (11)$$

where \mathbf{J} is defined in (5). To obtain an exponential convergence of the error:

$$\dot{\mathbf{e}} = -K\mathbf{e}, \quad K > 0, \quad (12)$$

plugging (12) into (11) yields the least square solution for the joint velocity:

$$\dot{\mathbf{q}} = \mathbf{J}^+ (\dot{\mathbf{x}}^* + K\mathbf{e}). \quad (13)$$

Here, the inverse kinematic problem has been solved by computing the pseudoinverse \mathbf{J}^+ of the Jacobian matrix [24]. Section III will introduce our alternative – parsimonious and QP-based – method.

III. PARSIMONIOUS TASK-SOLVING APPROACH

When the system is redundant, multiple joint configurations may satisfy the task while creating different behaviors. An explicit resolution of the inverse kinematic problem, when no constraint is specified, relies on the pseudoinverse \mathbf{J}^+ of the Jacobian, as in (13). It consists in minimizing the Euclidean norm of (3). However, this often requires restricting the admissible range of joint velocities. In particular, it does not account for the physical robot constraints (e.g. joint limits) nor for safety aspects (e.g. collisions). To consider these aspects, an optimization problem has to be formulated and solved by a numerical solver.

A quadratic programming resolution is the usual process to obtain the desired $\dot{\mathbf{q}}$. It minimizes the same 2-norm as the pseudoinverse but allows defining a set of constraints to be satisfied at any time:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|_2 \\ \text{s.t.} \quad & \mathbf{A}\dot{\mathbf{q}} = \mathbf{b}, \quad \mathbf{C}\dot{\mathbf{q}} \leq \mathbf{d} \end{aligned} \quad (14)$$

where \mathbf{A} , \mathbf{C} are the linear coefficients matrices and \mathbf{b} , \mathbf{d} the constant vectors in the equality and inequality constraints, respectively. Despite an advanced state of the art in the field, one drawback with (14) is the generation of joint velocities for which every component is non-null, regardless the number of DOF controlled in the task space. This minimizes the energy dissipated by the system but not the robot overall displacement.

In non-rigid industrial settings where the robot's free space can change, a parsimonious resolution would be more appropriate. Here, we aim at minimizing the number of joints involved in the task, thus reducing the robot motion.

The corresponding optimization problem is:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \|\dot{\mathbf{q}}\|_0 \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \end{aligned} \quad (15)$$

with $\|\dot{\mathbf{q}}\|_0$ the number of nonzero components in $\dot{\mathbf{q}}$. This is a NP-hard problem that requires a combinatorial approach and dual-arm complexity does not allow to solve it on line. It has been shown [25] that the sparsest solution can also be obtained by solving the following equivalent problem:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \|\dot{\mathbf{q}}\|_1 \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}. \end{aligned} \quad (16)$$

By easily transforming (16) into a linear program, a sparse solution can be efficiently generated.

Fuchs also proved [26] that a parametrized quadratic program, known in statistics as the lasso (Least Absolute Shrinkage and Selection Operator), and expressed as:

$$\min_{\dot{\mathbf{q}}} \quad \frac{1}{2} \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_1, \quad \lambda > 0 \quad (17)$$

converges to the same minimum point solution, i.e. having least l_1 -norm, when $\lambda = 0^+$. The advantage of using (17) instead of (16) is that the kinematic problem is formulated as

a cost function and not as an equality constraint. This leads to a softer behavior, since a solution is always found. In fact, the optimization process tries to minimize the tracking error of the task. If the kinematic problem is undetermined and several solutions exist, then the sparsest one is returned. Instead, with (16), no valid solution is computed when $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$ cannot be satisfied.

An algorithm for solving this optimization problem, while also including both equality and inequality constraints, has been proposed in [27]. The objective is to find a solution to the constrained lasso problem:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_1, \quad \lambda > 0 \\ \text{s.t.} \quad & \mathbf{A}\dot{\mathbf{q}} = \mathbf{b}, \quad \mathbf{C}\dot{\mathbf{q}} \leq \mathbf{d} \end{aligned} \quad (18)$$

In [27], the problem is reformulated as a standard QP: the l_1 penalty term $\|\dot{\mathbf{q}}\|_1$ can be handled by decomposing $\dot{\mathbf{q}}$ into its positive and negative parts, such as $\dot{\mathbf{q}} = \dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-$ with $\dot{\mathbf{q}}^+ \geq 0$, $\dot{\mathbf{q}}^- \geq 0$. Then, plugging $\|\dot{\mathbf{q}}\|_1 = \dot{\mathbf{q}}^+ + \dot{\mathbf{q}}^-$ in (18):

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \begin{pmatrix} \dot{\mathbf{q}}^+ \\ \dot{\mathbf{q}}^- \end{pmatrix}^T \begin{pmatrix} \mathbf{J}^T \mathbf{J} & -\mathbf{J}^T \mathbf{J} \\ -\mathbf{J}^T \mathbf{J} & \mathbf{J}^T \mathbf{J} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}^+ \\ \dot{\mathbf{q}}^- \end{pmatrix} \\ & + \left(\lambda \mathbf{1}_{2p} - \begin{pmatrix} \mathbf{J}^T \dot{\mathbf{x}} \\ -\mathbf{J}^T \dot{\mathbf{x}} \end{pmatrix} \right)^T \begin{pmatrix} \dot{\mathbf{q}}^+ \\ \dot{\mathbf{q}}^- \end{pmatrix} \\ \text{s.t.} \quad & (\mathbf{A} \quad -\mathbf{A}) \begin{pmatrix} \dot{\mathbf{q}}^+ \\ \dot{\mathbf{q}}^- \end{pmatrix} = \mathbf{b}, \\ & (\mathbf{C} \quad -\mathbf{C}) \begin{pmatrix} \dot{\mathbf{q}}^+ \\ \dot{\mathbf{q}}^- \end{pmatrix} \leq \mathbf{d}, \\ & \dot{\mathbf{q}}^+ \geq 0, \quad \dot{\mathbf{q}}^- \geq 0 \end{aligned} \quad (19)$$

which is the formulation of a standard QP of $2p$ variables, p being the dimension of \mathbf{q} .

IV. CONSIDERATIONS ON THE JOINT CONSTRAINTS

Taking into account joint limit constraints in dual-arm manipulation has been addressed in [12]. In that paper, joint limits avoidance is integrated in a prioritized hierarchy of tasks, with priority depending on whether the system is redundant or not. However, this architecture is based on the projection on the null space of higher priority tasks, and does not handle strict constraints. Moreover, it does not consider joint velocity limits.

Our approach to address joint limits is twofold:

- setting hard constraints at the joint velocity level for each task, to remain inside joint position and velocity ranges at anytime.
- creating a lower priority task in which a repulsive cost function $\dot{\mathbf{q}}_r$ pushes joints away from their limits.

A. Joint position and velocity limits

Let us define the lower and upper bounds on joint positions and velocities as:

$$\begin{aligned} \underline{\mathbf{q}} &\leq \mathbf{q} \leq \bar{\mathbf{q}}, \\ \underline{\dot{\mathbf{q}}} &\leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}}. \end{aligned} \quad (20)$$

Since the control variables are joint velocities, we want to express the position constraint as a velocity constraint. In a discrete control loop of sampling time $T > 0$, the joint velocity remains constant between two iterations. Thus, if k is the current iteration and the position constraint has to be satisfied at $k + 1$, this can be written as:

$$\underline{\mathbf{q}} \leq \mathbf{q}_k + \dot{\mathbf{q}}T \leq \bar{\mathbf{q}}, \quad (21)$$

which, plugged into (20), gives:

$$\begin{aligned} \frac{\underline{\mathbf{q}} - \mathbf{q}_k}{T} &\leq \dot{\mathbf{q}} \leq \frac{\bar{\mathbf{q}} - \mathbf{q}_k}{T}, \\ \underline{\dot{\mathbf{q}}} &\leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}}. \end{aligned} \quad (22)$$

From (22), we keep the most restrictive condition, to define the final lower and upper bounds, respectively $\underline{\dot{\mathbf{Q}}}$ and $\bar{\dot{\mathbf{Q}}}$:

$$\begin{aligned} \underline{\dot{\mathbf{Q}}} &= \max \left\{ \underline{\dot{\mathbf{q}}}, \frac{\underline{\mathbf{q}} - \mathbf{q}_k}{T} \right\} \\ \bar{\dot{\mathbf{Q}}} &= \min \left\{ \bar{\dot{\mathbf{q}}}, \frac{\bar{\mathbf{q}} - \mathbf{q}_k}{T} \right\}. \end{aligned} \quad (23)$$

B. Repulsive cost function for joint limits avoidance

Although a dual-arm robot is highly redundant with respect to a relative cooperative task, reaching joint position boundaries limits its operational capabilities. This may lead to suboptimal solutions, due to the reduction of the admissible solution space. To avoid this, we define a repulsive vector representing a target velocity for avoiding joint limits.

First, let $\alpha \in [0, 1]$ be a tunable parameter that sets the critical parts of the joint range $[q, \bar{q}]$ (i.e., the areas in which the repulsive action should be enabled). This leads to the definition of lower and upper activation thresholds (shown for joint i in Fig. 2), respectively:

$$\begin{aligned} \underline{q}^* &= \underline{q} + \alpha \frac{\bar{q} - \underline{q}}{2} \\ \bar{q}^* &= \bar{q} - \alpha \frac{\bar{q} - \underline{q}}{2}. \end{aligned} \quad (24)$$

If a joint position is inside $[\underline{q}^*, \bar{q}^*]$, no repulsive action is applied. In this case, the reference velocity for this joint is 0 (to avoid unnecessary motion). Otherwise, it is defined as a ratio of the minimum/maximum velocity that can be applied, depending whether the critical part is near the lower or upper joint limit. The repulsive value $\dot{q}_{r,i}$ associated to joint i is depicted in Fig. 2 and computed as:

$$\dot{q}_{r,i} = \begin{cases} 0, & q_i \in]\underline{q}_i^*; \bar{q}_i^*[\\ \frac{(q_i - \underline{q}_i^*)^2}{(\underline{q}_i - \underline{q}_i^*)^2} \bar{\dot{q}}_i & q_i \in [\underline{q}_i; \underline{q}_i^*] \\ \frac{(q_i - \bar{q}_i^*)^2}{(\bar{q}_i - \bar{q}_i^*)^2} \underline{\dot{q}}_i & q_i \in [\bar{q}_i^*; \bar{q}_i]. \end{cases} \quad (25)$$

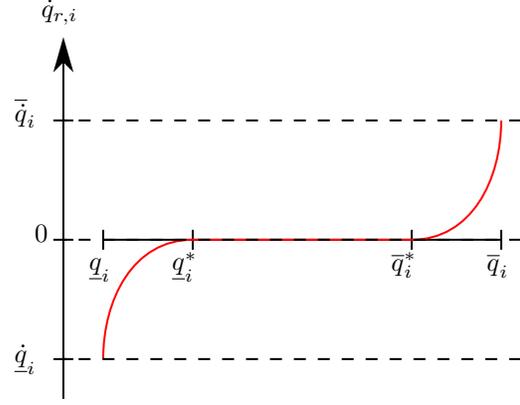


Fig. 2. Evolution of repulsive value associated to joint i with respect to joint position.

V. HIERARCHICAL SPARSE QUADRATIC PROGRAMMING ARCHITECTURE

The parsimonious task solving method and joint limits avoidance strategy are integrated in a hierarchical sparse quadratic programming framework.

The primary task is to ensure proper trajectory tracking of the desired $\dot{\mathbf{x}}^*$, while minimizing the generated motion and respecting joint limits. We denote as QP1 the corresponding optimization process and $\dot{\mathbf{q}}_1$ the generated solution:

$$\begin{aligned} \dot{\mathbf{q}}_1 &\in \min_{\dot{\mathbf{q}}} \frac{1}{2} \|\mathbf{J}\dot{\mathbf{q}} - (\dot{\mathbf{x}}^* + K\mathbf{e})\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_1, \quad \lambda = 0^+ \\ \text{s.t.} \quad &\underline{\dot{\mathbf{Q}}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{Q}}}. \end{aligned} \quad (26)$$

Here, the l_2 -norm term solves the closed-loop inverse kinematic problem presented in Section II-C and boundaries for $\dot{\mathbf{q}}$ are the ones defined in (23). The l_1 -norm penalty term λ is a real parameter such as $\lambda \in [0^+, \|J^T(\dot{\mathbf{x}}^* + K\mathbf{e})\|_\infty]$ in order to obtain a sparse solution. By taking λ as small as possible inside the permissible range provides the sparsest solution amongst those satisfying the task. Indeed, a greater value of λ increases the sparsity but also the tracking error.

The secondary task, for joint limits avoidance, is implemented as another QP (referred to as QP2). It provides an additional joint velocity vector $\dot{\mathbf{q}}_2$ solution of:

$$\begin{aligned} \dot{\mathbf{q}}_2 &\in \min_{\dot{\mathbf{q}}} \|\dot{\mathbf{q}} - \dot{\mathbf{q}}_1\|_2 \\ \text{s.t.} \quad &\underline{\dot{\mathbf{Q}}} - \dot{\mathbf{q}}_1 \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{Q}}} - \dot{\mathbf{q}}_1, \\ &\mathbf{J}\dot{\mathbf{q}} = \mathbf{0}. \end{aligned} \quad (27)$$

Adding the equality constraint $\mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$ restricts the solution space to the null space of the primary task, and does not affect its performance. Joint constraints have to be updated to take into account the velocity generated by $\dot{\mathbf{q}}_1$. Finally, the joint velocity sent to the robot is:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2. \quad (28)$$

The complete Hierarchical Sparse QP control framework is depicted in Fig. 3.

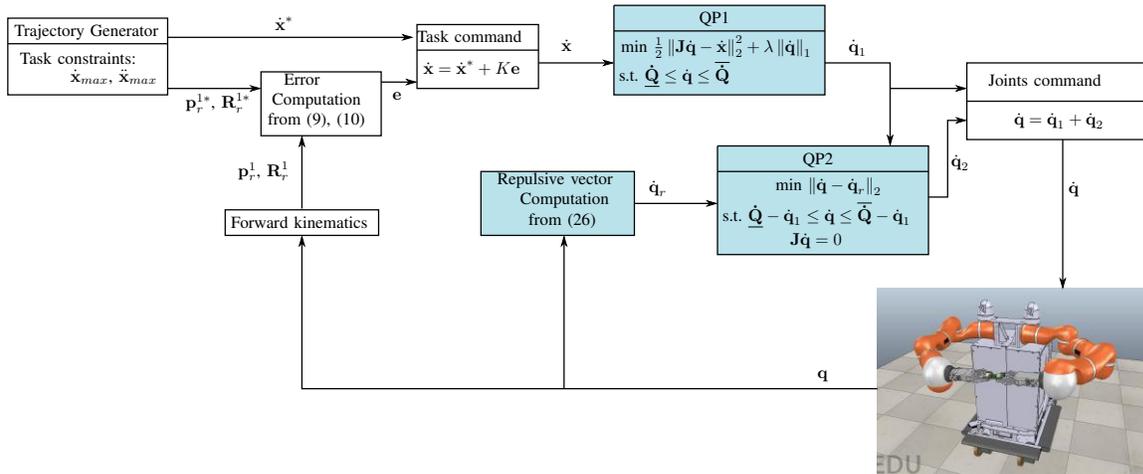


Fig. 3. The Hierarchical Sparse QP closed-loop kinematic scheme with trajectory generation.

VI. EXPERIMENTAL RESULTS

The efficiency and performance of the presented framework have been tested in simulation and on the real dual-arm cobot BAZAR designed at LIRMM. BAZAR is equipped with two 7-DOF Kuka LWR4 arms and an omni-directional mobile base (fixed in this study).

All experiments are performed on a computer with Intel(R) Core(TM) i7-6600U CPU running Linux. Our approach has been implemented in C++ using the Knowbotics framework developed at LIRMM. Simulation scenarios are performed on the Virtual Robot Experimentation Platform (V-REP)². The Fast Research Interface Library (FRI)³ was used to communicate with the Kuka arms during the real experiment, and the controller sample time was set to $T = 5$ ms.

The Reflexxes Motion Library [28] is used as trajectory generator for \mathbf{x}^* and $\dot{\mathbf{x}}^*$, given a set of constraints.

We use a gain $K = 10$ to compensate the tracking error. The l_1 -norm penalization factor is set to $\lambda = 1 \times 10^{-6}$, and $\alpha = 0.1$ is taken for the repulsive cost function. Joint limits used are the nominal LWR4 ones.

Video of the experiments is available at <http://bit.do/esfyu>.

A. Assembly task scenario

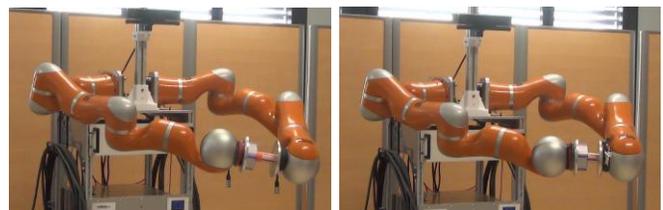
Our control scheme has been validated on the real robot through an assembly task in which a peg-in-hole operation is simulated. A peg is attached to one arm while a shape with corresponding hole is fixed to the other. Screwing motion is also required to realize the insertion. The task is divided into two parts:

- At the beginning, the two arms are located far from each other as in Fig. 1. The first relative motion aims at bringing the objects closer and aligning the peg with the hole, as shown in the left pictures of Fig. 4.
- then, the screwing operation is performed to insert the peg in the hole.

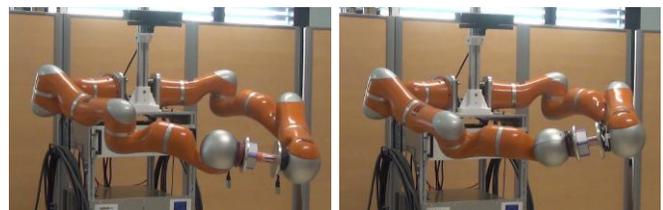
The scenario is performed with three different task-solving configurations: regular QP from (14), sparse QP (QP1), and hierarchical sparse QP (QP1 + QP2). Hard constraints on joint limits are integrated in any case.



(a) Standard QP



(b) Sparse QP



(c) Hierarchical Sparse QP

Fig. 4. Initial and final configuration reached during the simulated assembly task using the different task-solving configurations.

B. Results

From a high-level point of view, initial and final configurations obtained for the screwing task reveal the behavioral differences between the methods in terms of generated motions. The screwing operation, that could be intuitively performed by a simple action, generated a high occupancy of

²<http://www.coppeliarobotics.com/>

³<https://cs.stanford.edu/people/tkr/fri/html/>

the workspace using the standard QP (Fig. 4(a)). Conversely, the parsimonious solution has resulted in local and economic movements as demonstrated by Fig. 4(b) that shows similar intermediate and final configurations, except for the grippers' orientations. In the last case (Fig. 4(c)), the secondary task has slightly modified the configurations to avoid limits for the right shoulder and the wrists.

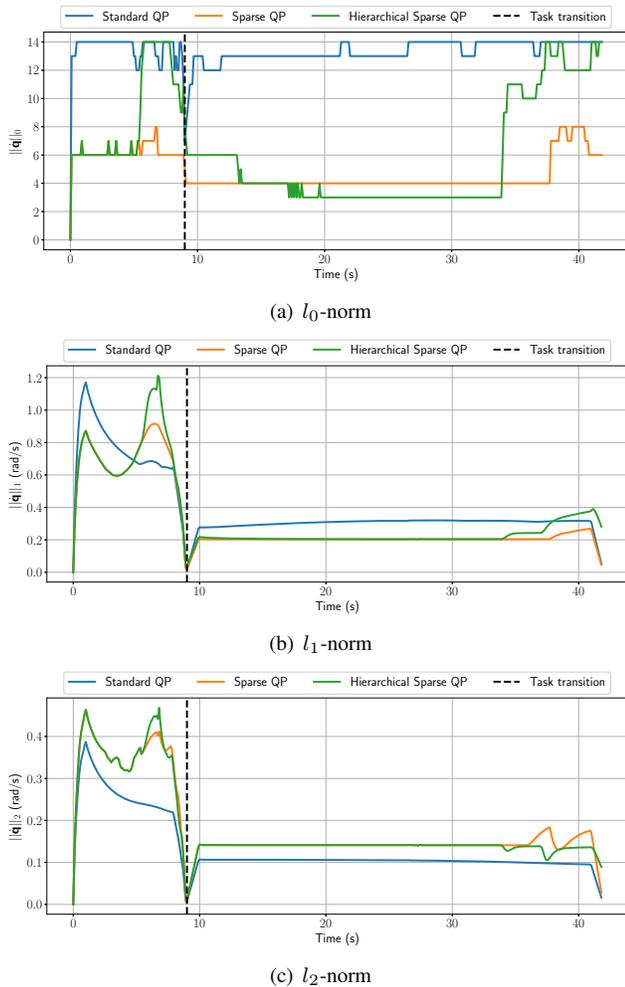


Fig. 5. Comparison of joint velocity norms during the assembly task execution. A threshold value of $1 \times 10^{-3} \text{ rad s}^{-1}$ is taken to differentiate active from inactive joints (l_0 -norm).

TABLE I

COMPARISON OF THE METHODS DURING THE ASSEMBLY TASKS USING DIFFERENT METRICS.

Metric	Standard QP	Sparse QP	Hierarchical Sparse QP
$\int_0^\infty \ \dot{\mathbf{q}}\ _0 dt$	13.385	4.782	6.311
$\int_0^\infty \ \dot{\mathbf{q}}\ _1 dt$	0.393	0.308	0.333
$\int_0^\infty \ \dot{\mathbf{q}}\ _2 dt$	0.133	0.184	0.182

Numerical results from Fig. 5 and I confirm this observation. The overall number of actuated joints as well as the sum of absolute joint velocities have been reduced with the parsimonious approach. Indeed, aligning both end-effectors

from an arbitrary configuration required the use of six joints for the most part, corresponding to the number of controlled task variables. Even less joints are needed to execute the screwing movement, because of the geometric properties of the operation. As expected, the standard QP approach minimizes the energy consumption during the task. However, this resolution involves the use of almost every joints, regardless the geometric complexity of the task. When the joint limits repulsive cost function is enabled (Hierarchical Sparse QP), additional joints are temporarily activated to leave critical parts of position ranges: between 5.4s and 9s the right shoulder approaches a boundary, and from 34s to the end of the task, the two wrists reach the repulsive area. However, as seen in Fig. 5(a) and Fig. 5(b), this feature has only local effect on the robot's behavior and the hierarchical sparse QP still provides much sparser motions than the standard one, as demonstrated in Table I. Note that the hierarchical sparse QP may give a smaller number of actuated joints compared to the sparse QP (as is the case in the middle part of the task). This is due to the algorithms which are local and do not ensure global optimality.

VII. CONCLUSION AND FUTURE WORK

This paper presented a new kinematic control strategy for cooperative dual manipulator robots. Based on a hierarchy of QP, the obtained solution is one that activates the minimal number of joints to perform coordinated motions, while avoiding joint limits. Experiments on a real dual-arm robot revealed the benefits of our task-solving method compared to the standard QP. The resulting economic behavior is an appealing feature for numerous robotic applications.

Future work will focus on addressing parsimony in simultaneous local and global manner by mixing kinematic and impedance control for improving the interaction with human operator and robot in the collaborative manipulation tasks.

ACKNOWLEDGMENT

This research has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement no 731330 (VERSATILE: Innovative robotic applications for highly reconfigurable production lines).

REFERENCES

- [1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation survey," *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [2] P. Chiacchio and S. Chiaverini, *Complex Robotic Systems*. Springer-Verlag Berlin Heidelberg, 1998, ch. 1, pp. 1–31.
- [3] J. Felip and A. Morales, "A solution for the cap unscrewing task with a dual arm sensor-based system," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.
- [4] M. P. Polverini, A. M. Zanchettin, S. Castello, and P. Rocco, "Sensorless and constraint based peg-in-hole task execution with a dual-arm robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [5] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," *Journal of dynamic systems, measurement, and control*, vol. 118, no. 4, pp. 691–697, 1996.

- [6] D. Almeida, F. E. Vina, and Y. Karayiannidis, "Bimanual folding assembly: Switched control and contact point estimation," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016.
- [7] W. Owen, E. Croft, and B. Benhabib, "Stiffness optimization for two-armed robotic sculpting," *Industrial Robot: An International Journal*, vol. 35, no. 1, pp. 46–57, 2008.
- [8] S. Ahmad and S. Luo, "Coordinated motion control of multiple robotic devices for welding and redundancy coordination through constrained optimization in cartesian space," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 409–417, 1989.
- [9] A. Mohri, M. Yamamoto, and G. Hirano, "Cooperative path planning for two manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1996.
- [10] M. Faroni, M. Beschi, A. Visioli, and L. M. Tosatti, "A global approach to manipulability optimisation for a dual-arm manipulator," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016.
- [11] Y. Hu, B. Huang, and G.-Z. Yang, "Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [12] D. Ortenzi, R. Muthusamy, A. Freddi, A. Monteriù, and V. Kyrki, "Dual-arm cooperative manipulation under joint limit constraints," *Robotics and Autonomous Systems*, vol. 99, pp. 110–120, 2018.
- [13] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.
- [14] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [15] M. Yazdani, "Fast human movements and sparse optimal control policies," Ph.D. dissertation, University of California, San Diego, 2012.
- [16] J.-P. Gauthier, B. Berret, and F. Jean, "A biomechanical inactivation principle," *Proceedings of the Steklov Institute of Mathematics*, vol. 268, no. 1, pp. 93–116, 2010.
- [17] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
- [18] C. L. Lewis and A. A. Maciejewski, "Trajectory generation for cooperating robots," in *IEEE International Conference on Systems Engineering*, 1990.
- [19] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Task-oriented kinematic control of two cooperative 6-dof manipulators," in *IEEE American Control Conference*, 1993.
- [20] B. Cao, G. I. Dodds, and G. W. Irwin, "Redundancy resolution and obstacle avoidance for cooperative industrial robots," *Journal of Field Robotics*, vol. 16, no. 7, pp. 405–417, 1999.
- [21] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [22] R. S. Jamisola and R. G. Roberts, "A more compact expression of relative jacobian based on individual manipulator jacobians," *Robotics and Autonomous Systems*, vol. 63, pp. 158–164, 2015.
- [23] J. Luh, M. Walker, and R. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.
- [24] A. Albert, *Regression and the Moore-Penrose pseudoinverse*. Elsevier, 1972.
- [25] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2558–2567, 2002.
- [26] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE transactions on Information theory*, vol. 50, no. 6, pp. 1341–1344, 2004.
- [27] B. R. Gaines and H. Zhou, "Algorithms for fitting the constrained lasso," *arXiv preprint arXiv:1611.01511*, 2016.
- [28] T. Kröger, "Opening the door to new sensor-based robot application—the reflexes motion libraries," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.