2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Distributed Processing Approach for Smooth Task Transitioning in Strict Hierarchical Control

Francesco Tassi[1,2] and Arash Ajoudani[1]

***Abstract*— To enhance robots' applicability in real-world scenarios, it is essential to establish a complex and multi-tasking behaviour, inspired by human nature. To this purpose, from a hardware perspective, a high number of degrees of freedom is necessary, as is the case for humanoids and collaborative mobile manipulators. From a software standpoint instead, complex hierarchical strategies are often used to define a set of behaviours that the robot should reflect in strict hierarchical order. Their main issue however, is related to the lack of continuity when their stack of tasks is changed. Existing works that address this issue clearly present a trade-off between optimality assurance during transition and computational costs. Here, we employ a distributed processing approach that enables not only the minimization of computational costs, but also continuous optimality and constraints feasibility even under sharp transitions. The approach is tested during three task transitions, for different tasks such as constrained trajectory tracking, obstacle avoidance, and postural optimization. Two mobile manipulators are used, each having 10 DoF, and the results confirm the smoothness of the generated solutions.**

I. Introduction

Hierarchical control has been deeply studied for highly redundant robots such as humanoids [1] or dual-arm space robots [2], but also for simpler animal-shaped [3] or Collaborative Robots [4]–[6]. Aiming towards a more synergistic and widespread employment of robots in real-life environments, some of the most limiting factors are related to robots' autonomy [7] and their interaction with the environment and humans [8]. In this sense, hierarchical control is essential to fully exploit robots' potentials and allow the formulation of multiple objectives that are accomplished hierarchically.

Within hierarchical control, a distinction is made between *non-strict* and *strict* hierarchies. The first one consists in the relative weighting between different tasks [9], [10], which however does not ensure that the task holding higher priority is strictly enforced. This is achieved instead through the latter category, where the secondary tasks are projected into the null-space of the primary ones, ensuring a *strict* hierarchy of tasks. In particular, Hierarchical Quadratic Programming (HQP) [11], [12] is the most used strategy to formulate a *strict* Stack of Tasks (SoT) [13]. However, one of the most common limitations of HQP is related to the problem of task transitioning. When the priority of any task is changed, or whether a task is added or removed, a discontinuity is generated in the optimal solution [14]. Multiple studies have addressed this issue, and will be further analysed in Sec. II. The layout of the paper is as follows. First, in Sec. II the background to the problem of smooth task transitioning is provided, assessing current limitations. In Sec. III the proposed method is formalised. Sec. IV reports the experimental validations, and Sec. V provides final conclusions.

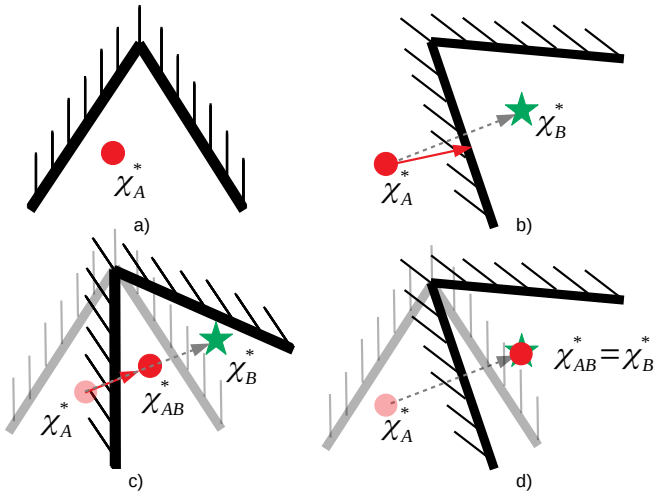

Fig. 1: Gradual constraints shift. Changing from a) to b) leads to infeasibility with existing approaches. The red dot and the green star are the current and final optimum, respectively. With the proposed approach (from a to c to d), we decouple the instantaneous solution from the final one, and enable gradual constraints' variation, allowing the intermediate solutions to remain in the feasibility range.

II. Related Works

When the SoT remains unaltered, the hierarchical solution is ensured to be continuous, even in case of ill-conditioning [11], [15]. However, discontinuities arise during the transition between different tasks, and multiple methods have been studied to this purpose [16]. Generally, the easiest solution is to use weight-based methods, which employ a series of smoothly-varying weights to ensure continuity [17]. However, all the tasks must be known a priori and cannot be changed online. In [18], an intermediate value approach is developed during the transition period, with the issue of being non-optimal since this is not written as an optimization problem, hence the motion is not guaranteed to be feasible (given the lack of constraints). These type of approaches interpolate between the *pre-* and *post*-transition solutions, hence, besides the aforementioned problem of non-optimality and potential infeasibility, there exists the issue of increased computational times due to the multiple optimizations.

[1] Human-Robot Interfaces and Interaction lab., Istituto Italiano di Tecnologia, Genoa, Italy, Email: Francesco.Tassi@iit.it
[2] RAISE Ecosystem, Genoa, Italy.
This work was carried out within the framework of the project "RAISE - Robotics and AI for Socio-economic Empowerment" and has been supported by European Union - NextGenerationEU. However, the views and opinions expressed are those of the authors alone and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

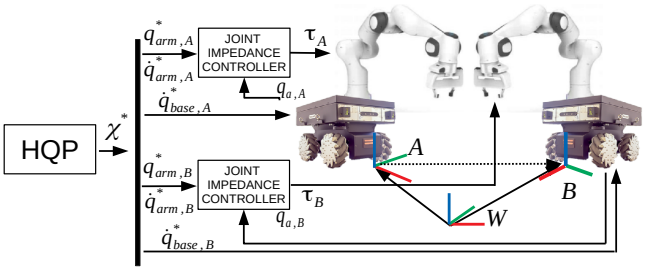

Fig. 2: Layout of the overall multi-robot system. Two identical mobile manipulators are employed, with A and B local frames respectively. All the variables in the HQP problem are expressed in world frame W.

The authors in [19] propose a method that does not increase computational costs by *softening* the hierarchical problem during the transition phase. However, again the issue of optimality during the transition recurs. Indeed, by *softening* the problem, *strict* hierarchical prioritization is lost. Besides, its formulation becomes intricate as the number of task swaps increases and cannot handle inequality constraints. In [20] a generalized hierarchical control acts on the activation parameters of the generalized projector, achieving a smooth transition. However, the feasibility of the solution is not always ensured since it depends from the robot's state. Recently, both [21] and [22] proposed a continuous task transitioning approach for HQP, which maintains the optimality during the transition. However, they suffer from the increase in computational times and the necessity of successive activations with multiple gains.

A. Contributions

As mentioned above, the current works in smooth hierarchy transitioning suffer from a clear trade-off between either the absence of optimality enforced during the transition phase, or the increase in computational times. We aim to address both issues with a distributed formulation that keeps minimum computational times while constantly ensuring *strict* optimality during the smooth transition.

Indeed, we propose a distributed processing technique that exploits the computational capacities of the multi-DoF systems involved in the task (two mobile manipulators in our study). This approach is scalable, and can be extended to applications like large-scale multi-robot systems and swarm robotics, where often not all available robots are actively employed in the computation. In simpler cases, a single robot with two available CPUs (onboard or offboard) can also exploit our proposed approach.

The idea is to find the solution for the new optimization problem, and use it to smoothly drive the old solution to the new one. This distributed approach enables to change not only the SoT but also constraints entirely, without potential infeasibility issues deriving from a sudden change in constraints that would occur from a non-distributed approach. Fig. 1 shows this concept, where a) is the initial configuration and the red dot indicates the current solution. Upon constraints' change (b), the new solution (green star) lies outside of the feasibility range, and the constraints on

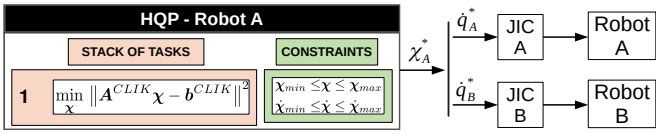

Fig. 3: Block scheme of the problem solved by robot A when no transition has been triggered. The output is $\boldsymbol{\chi}_A^*$ and includes the joint velocities for both robots $\dot{\boldsymbol{q}}_A^*, \dot{\boldsymbol{q}}_B^*$.

velocity/acceleration do not allow the red dot to move fast enough (red arrow), thus leading to infeasibility. With the proposed approach, it is possible to decouple the instantaneous solution from the final one, thus enabling gradual constraints' variation, allowing the intermediate solutions to remain in the feasibility range, as occurs from a) to c) to d) (where the transparent configuration is the initial one).

We highlight here the overall proposed contributions with respect to the existing approaches:

- Solution's optimality and constraints' satisfaction are ensured at any stage during the transition.
- Guaranteed feasibility in case of sudden change in constraints (Fig. 1).
- Minimal computational cost due to the distributed processing approach.
- No need for sequential tasks insertion/removal, which requires intricate structure, numerous weights and increased computational costs.

III. Methodology

A. Hierarchical Quadratic Programming Problem

Through a HQP-based control, the strict optimization problem for two successive priorities writes:

$$\min_{\boldsymbol{\chi}} \ \frac{1}{2} \left\| \boldsymbol{A^1}\boldsymbol{\chi} - \boldsymbol{b^1} \right\|^2 \tag{1}$$

$$s.t. \ \ \boldsymbol{C}_1^1\boldsymbol{\chi} \leq \boldsymbol{d}_1^1, \ \ \ldots, \ \ \boldsymbol{C}_{n_{i_1}}^1\boldsymbol{\chi} \leq \boldsymbol{d}_{n_{i_1}}^1 \tag{2}$$

$$\boldsymbol{E}_1^1\boldsymbol{\chi} = \boldsymbol{f}_1^1, \ \ \ldots, \ \ \boldsymbol{E}_{n_{e_1}}^1\boldsymbol{\chi} = \boldsymbol{f}_{n_{e_1}}^1 \tag{3}$$

and the second level, with lower priority, as:

$$\min_{\boldsymbol{\chi}} \ \frac{1}{2} \left\| \boldsymbol{A^2}\boldsymbol{\chi} - \boldsymbol{b^2} \right\|^2 \tag{4}$$

$$s.t. \ \ \boldsymbol{C}_1^2\boldsymbol{\chi} \leq \boldsymbol{d}_1^2, \ \ \ldots, \ \ \boldsymbol{C}_{n_{i_2}}^2\boldsymbol{\chi} \leq \boldsymbol{d}_{n_{i_2}}^2 \tag{5}$$

$$\boldsymbol{E}_1^2\boldsymbol{\chi} = \boldsymbol{f}_1^2, \ \ \ldots, \ \ \boldsymbol{E}_{n_{e_2}}^2\boldsymbol{\chi} = \boldsymbol{f}_{n_{e_2}}^2 \tag{6}$$

$$\boldsymbol{A}_1\boldsymbol{\chi} = \boldsymbol{A}_1\boldsymbol{\chi}_1^*, \tag{7}$$

where $\boldsymbol{\chi} \in \mathbb{R}^s$ is the optimization variable, and the optimality condition defined between successive Quadratic Programming (QP) solutions in (7) ensures *strict* priorities [23].

Considering $N_t \in \mathbb{N}$ as the total number of tasks, and $k \in \{1, \ldots N_t\}$ as the priority levels with decreasing importance from 1 (highest) to N_t (lowest), the generic formulation for the $k-$th generic task becomes:

$$\min_{\boldsymbol{\chi}} \ \frac{1}{2} \left\| \boldsymbol{A}^k\boldsymbol{\chi} - \boldsymbol{b}^k \right\|^2 \tag{8}$$

$$s.t. \ \ \boldsymbol{C}_1^k\boldsymbol{\chi} \leq \boldsymbol{d}_1^k, \ \ \ldots, \ \ \boldsymbol{C}_{n_{i_k}}^k\boldsymbol{\chi} \leq \boldsymbol{d}_{n_{i_k}}^k \tag{9}$$

$$\boldsymbol{E}_1^k\boldsymbol{\chi} = \boldsymbol{f}_1^k, \ \ \ldots, \ \ \boldsymbol{E}_{n_{e_k}}^k\boldsymbol{\chi} = \boldsymbol{f}_{n_{e_k}}^k \tag{10}$$

$$\boldsymbol{A}^1\boldsymbol{\chi} = \boldsymbol{A}^1\boldsymbol{\chi}_1^*, \ \ \ldots, \ \ \boldsymbol{A}^{k-1}\boldsymbol{\chi} = \boldsymbol{A}^{k-1}\boldsymbol{\chi}_{k-1}^*. \tag{11}$$

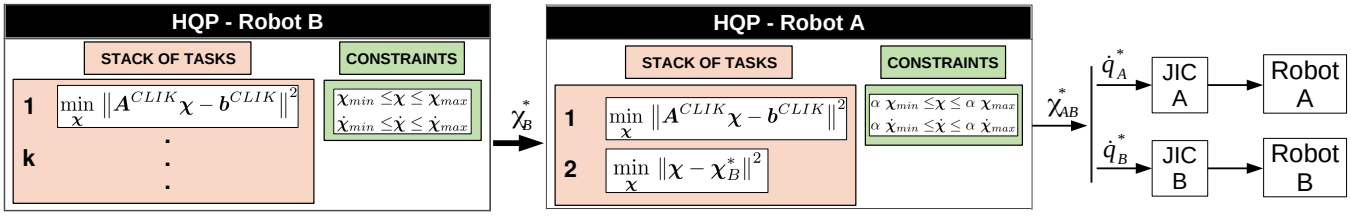

Fig. 4: Block scheme of the distributed task transitioning approach. Robot B is in charge of computing the (discontinuous) solution to the new problem $\boldsymbol{\chi}_B^*$, which is then fed to Robot A that is responsible for smoothing this transition via $\boldsymbol{\chi}_{AB}^*$. Both solutions have the same structure, that includes joint velocities for both robots $\dot{\boldsymbol{q}}_A^*, \dot{\boldsymbol{q}}_B^*$ as well.

Here, t_k is the dimension of each k-th task, subject to the k-th number of equality and inequality constraints in $n_{e_k} \in \{n_{e_1}, \dots n_{e_{Nt}}\}$ and $n_{i_k} \in \{n_{i_1}, \dots n_{i_{Nt}}\}$ respectively, with the dimensionality of each equality/inequality constraint at each k-th level being e_k and i_k, respectively. This leads to the following matrix dimensions:

$$
\begin{aligned}
\boldsymbol{A}^k &\in \{\boldsymbol{A}^1 \in \mathbb{R}^{t_1 \times s},\ \dots,\ \boldsymbol{A}^{N_t} \in \mathbb{R}^{t_{N_t} \times s}\} \\
\boldsymbol{b}^k &\in \{\boldsymbol{b}^1 \in \mathbb{R}^{t_1},\ \dots,\ \boldsymbol{b}^{N_t} \in \mathbb{R}^{t_{N_t}}\} \\
\boldsymbol{C_k} &= [\boldsymbol{C}_1^k \in \mathbb{R}^{i_1 \times s\ \ T},\ \dots,\ \boldsymbol{C}_{n_{i_k}}^k \in \mathbb{R}^{i_k \times s\ \ T}]^T \\
\boldsymbol{E_k} &= [\boldsymbol{E}_1^k \in \mathbb{R}^{e_1 \times s\ \ T},\ \dots,\ \boldsymbol{E}_{n_{e_k}}^k \in \mathbb{R}^{e_k \times s\ \ T}]^T \\
\boldsymbol{d_k} &= [\boldsymbol{d}_1^k \in \mathbb{R}^{i_1\ \ T},\ \dots,\ \boldsymbol{d}_{n_{i_k}}^k \in \mathbb{R}^{i_k\ \ T}]^T \\
\boldsymbol{f_k} &= [\boldsymbol{f}_1^k \in \mathbb{R}^{e_1\ \ T},\ \dots,\ \boldsymbol{f}_{n_{e_k}}^k \in \mathbb{R}^{e_k\ \ T}]^T,
\end{aligned}
$$

which allows to write a more compact notation

$$
\min_{\boldsymbol{\chi}} \ \frac{1}{2} \left\| \boldsymbol{A}^k \boldsymbol{\chi} - \boldsymbol{b}^k \right\|^2 \tag{12}
$$

$$
s.t.\ \ \boldsymbol{C_k} \boldsymbol{\chi} \leq \boldsymbol{d_k} \tag{13}
$$

$$
\boldsymbol{E_k} \boldsymbol{\chi} \leq \boldsymbol{f_k} \tag{14}
$$

$$
\boldsymbol{A}^1 \boldsymbol{\chi} = \boldsymbol{A}^1 \boldsymbol{\chi}_1^*,\ \dots,\ \boldsymbol{A}^{k-1} \boldsymbol{\chi} = \boldsymbol{A}^{k-1} \boldsymbol{\chi}_{k-1}^*. \tag{15}
$$

that includes the multiple equality/inequality constraints per each k-th level.

B. Stack of Tasks

a) Closed-Loop Inverse Kinematics: a Closed-loop Inverse Kinematics (CLIK) scheme is written in QP form as

$$
\min_{\dot{\boldsymbol{q}}} \ \| \boldsymbol{J} \dot{\boldsymbol{q}} - (\dot{\boldsymbol{x}}^d + \boldsymbol{K_p} (\boldsymbol{x}^d - \boldsymbol{x}_a)) \|^2 \tag{16}
$$

where $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the joint velocities and $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix. This closes the loop on the position error between the desired and actual position of the EE $\boldsymbol{x}_d$ and $\boldsymbol{x}_a \in \mathbb{R}^m$ respectively, whereas $\boldsymbol{K_p} \in \mathbb{R}^{m \times m}$ is the positive-definite diagonal gain matrix representing the Cartesian stiffness. Having to enforce the CLIK task for both robots, we define the optimization variable as

$$
\boldsymbol{\chi} = \begin{bmatrix} \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} \in \mathbb{R}^{s=2n} \tag{17}
$$

where $\dot{\boldsymbol{q}}_A, \dot{\boldsymbol{q}}_B \in \mathbb{R}^n$ are the the joint velocities of each robot. Therefore, the overall CLIK task can be written as

$$
\begin{aligned}
\min_{\boldsymbol{\chi}} \ & \| \boldsymbol{J}_A \dot{\boldsymbol{q}}_A - (\dot{\boldsymbol{x}}_A^d + \boldsymbol{K_{p_A}} (\boldsymbol{x}_A^d - \boldsymbol{x}_{a_A})) \|^2 + \\
& \| \boldsymbol{J}_B \dot{\boldsymbol{q}}_B - (\dot{\boldsymbol{x}}_B^d + \boldsymbol{K_{p_B}} (\boldsymbol{x}_B^d - \boldsymbol{x}_{a_B})) \|^2
\end{aligned} \tag{18}
$$

with all coordinates expressed in world frame (Fig. 2).

Referring to the generic (12), the CLIK tasks lead to

$$
\boldsymbol{A}^{CLIK} = \begin{bmatrix} \boldsymbol{J}_A & \boldsymbol{0}_{n \times n} \\ \boldsymbol{0}_{n \times n} & \boldsymbol{J}_B \end{bmatrix}, \tag{19}
$$

$$
\boldsymbol{b}^{CLIK} = \begin{bmatrix} \dot{\boldsymbol{x}}_A^d + \boldsymbol{K_{p_A}} (\boldsymbol{x}_A^d - \boldsymbol{x}_{a_A}) \\ \dot{\boldsymbol{x}}_B^d + \boldsymbol{K_{p_B}} (\boldsymbol{x}_B^d - \boldsymbol{x}_{a_B}) \end{bmatrix}. \tag{20}
$$

This clearly constitutes the most important task in each of the SoTs under study, and is therefore always maintained as the first hierarchical layer, as shown in Fig.s 3 and 4.

b) Arm Postural Task: The goal is to drive the current joints' state to a desired one (e.g. mid-range of the robot's actuators). Based on the nature of the mobile manipulators used, the following state variable decomposition applies:

$$
\boldsymbol{\chi} = \begin{bmatrix} \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_{base,A} \\ \dot{\boldsymbol{q}}_{arm,A} \end{bmatrix} \\ \\ \begin{bmatrix} \dot{\boldsymbol{q}}_{base,B} \\ \dot{\boldsymbol{q}}_{arm,B} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_{base,A} \\ \dot{\boldsymbol{q}}_{arm,A} \end{bmatrix} \\ \\ \begin{bmatrix} \dot{\boldsymbol{x}}_{base,B} \\ \dot{\boldsymbol{q}}_{arm,B} \end{bmatrix} \end{bmatrix} \tag{21}
$$

in which $\dot{\boldsymbol{q}}_{base} \in \mathbb{R}^3$ and $\dot{\boldsymbol{q}}_{arm} \in \mathbb{R}^7$ are the mobile base and arm respective components. In particular, $\dot{\boldsymbol{x}}_{\boldsymbol{base}} = (\dot{x}_{base}, \dot{y}_{base}, \dot{\theta}_{base})$ is composed by the linear and angular velocities of the mobile base in the horizontal plane.

Being interested in the formulation of an arm-only postural task, this can be written in QP form as

$$
\min_{\boldsymbol{\chi}} ||\boldsymbol{q}_{arm} - \boldsymbol{q}_{arm}^d||^2 = \min_{\dot{\boldsymbol{q}}} ||\Delta t \dot{\boldsymbol{q}}_{arm} + \boldsymbol{q}_{arm}^{t-\Delta t} - \boldsymbol{q}_{arm}^d||^2,
$$

where $\boldsymbol{q}_{arm}^d \in \mathbb{R}^7$ is the desired joint position, Δt is the control period and $\boldsymbol{q}_{arm}^{t-\Delta t} \in \mathbb{R}^7$ is the optimal joint position obtained at the previous time step via Euler integration.

Overall, referred to (12), we achieve

$$
\boldsymbol{A}_A^{pos} = \Delta t\ \boldsymbol{\Gamma} \tag{22}
$$

$$
\boldsymbol{b}_A^{pos} = \boldsymbol{q}_{arm}^{t-\Delta t} - \boldsymbol{q}_{arm,A}^d \tag{23}
$$

$$
\boldsymbol{A}_B^{pos} = \Delta t\ \boldsymbol{\Gamma} \tag{24}
$$

$$
\boldsymbol{b}_B^{pos} = \boldsymbol{q}_{arm}^{t-\Delta t} - \boldsymbol{q}_{arm,B}^d \tag{25}
$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{7 \times n}$ is the selection matrix for the arm elements only. We employ this task in the experiments, to highlight the smoothness in task transition when it is activated.

c) Mobile Base Task: Based on (21), a similar task is defined for the mobile base elements only, to set a reference for $\boldsymbol{x}_{base}$, useful for obstacle avoidance, as:

$$
\min_{\boldsymbol{\chi}} ||\boldsymbol{x}_{base} - \boldsymbol{x}_{base}^d||^2, \tag{26}
$$

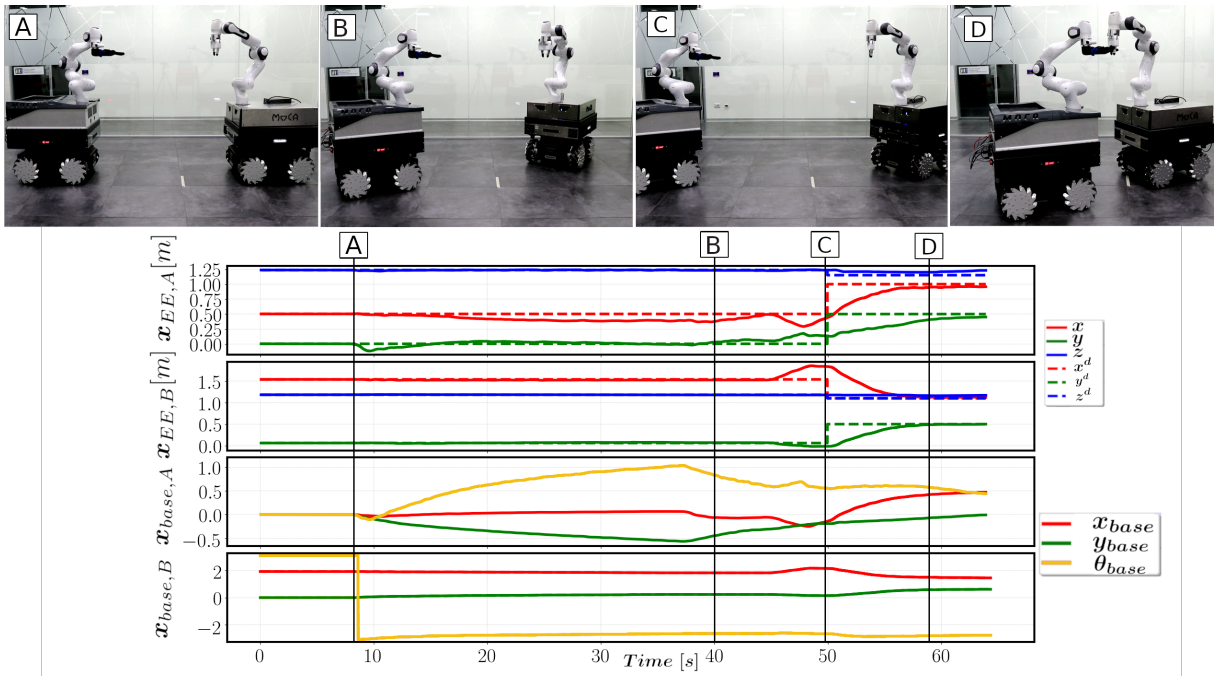

Fig. 5: Complete task sequence. Snapshots are taken at near to stationary conditions, after each task is accomplished. In particular: (A) initial conditions/activation of *SoT B1*; (B) *SoT B1* is accomplished and *SoT B2* is activated; (C) *SoT B2* is accomplished and *SoT B3* is activated, (D) final configuration. The plots show (top to bottom): both EE's actual $\boldsymbol{x}_a$ against desired $\boldsymbol{x}_d$ trajectories, and the measured positions of each mobile base's center of mass $\boldsymbol{x}_{base}$.

where $\boldsymbol{x}^d_{base} \in \mathbb{R}^3$ is the desired Cartesian position of the mobile base frame. Finally,

$$\boldsymbol{A}^{base}_A = \Delta t \ \boldsymbol{\Phi} \tag{27}$$

$$\boldsymbol{b}^{base}_A = \boldsymbol{x}^{t-\Delta t}_{base,A} - \boldsymbol{x}^d_{base,A} \tag{28}$$

$$\boldsymbol{A}^{base}_B = \Delta t \ \boldsymbol{\Phi} \tag{29}$$

$$\boldsymbol{b}^{base}_B = \boldsymbol{x}^{t-\Delta t}_{base,B} - \boldsymbol{x}^d_{base,B} \tag{30}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{3\times n}$ is the selection matrix for the mobile base elements $\boldsymbol{\Phi} = diag\{1,1,1\}$ and $\boldsymbol{x}^{t-\Delta t}_{base} \in \mathbb{R}^3$ is the optimal base position obtained at the previous time step via Euler integration.

C. Constraints

The constraints defined in the HQP are expressed at joint position, velocity, and acceleration levels, and are present at each $k-$th level of the SoT, for both robots:

$$\begin{aligned} \boldsymbol{q}_{min} \leq \boldsymbol{q}_{t-\Delta t} + \dot{\boldsymbol{q}} \ \Delta t \leq \boldsymbol{q}_{max} \\ \dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{max} \\ \ddot{\boldsymbol{q}}_{min} \leq \frac{\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{t-\Delta t}}{\Delta t} \leq \ddot{\boldsymbol{q}}_{max} \end{aligned} \tag{31}$$

where $\boldsymbol{q}_{min}, \dot{\boldsymbol{q}}_{min}, \ddot{\boldsymbol{q}}_{min} \in \mathbb{R}^n$ and $\boldsymbol{q}_{max}, \dot{\boldsymbol{q}}_{max}, \ddot{\boldsymbol{q}}_{max} \in \mathbb{R}^n$ are the minimum and maximum joint limits. For simplicity, we consider in the remainder all the constraints at position, velocity and acceleration levels through the notation

$$\boldsymbol{\chi}_{min} \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_{max}. \tag{32}$$

D. Distributed Hierarchical Framework

We analyse the scenario in which two mobile manipulators are available, but clearly this is scalable to larger multi-robot systems or even simpler robots, thus finding potential applications in the fields of large-scale multi-robot systems and swarm robotics, as mentioned in Sec. II.

The overall framework is shown in Fig. 2, where the two mobile manipulators are commanded via the optimal solutions obtained from the HQP. Two Joint Impedance Controllers (JIC) are used to obtain the torques for the manipulators, as it will be described in Sec. III-F. The goal is to overcome some of the most limiting aspects of the existing formulations currently developed for smooth task transitioning in hierarchical control, as already discussed in Sec. II. In particular, during the switching phase, the second robot is exploited to compute the solution of the new SoT. This allows to i) swap, insert or remove any number of tasks, without having to sequentially activate/deactivate them to ensure smoothness ii) potentially offload the computational burden to multiple robots, since in most smooth transitioning frameworks the optimization has to run multiple times, based on the number of tasks in transition. Hence, with the proposed strategy, not only a single optimization is only needed for each robot regardless of the number of tasks in transition (i), but also the computation time is offloaded to the robots available (ii).

As outlined in Fig. 3, initially (no task transitions) only robot A is responsible for solving the HQP problem, whose optimal solution $\boldsymbol{\chi}^*_A$ is passed to both robots via the JIC. As the transition phase is triggered, the proposed framework is activated (see Fig. 4) and the robot B will start calculating the new HQP problem with the renewed SoT and constraints. As soon as the new (discontinuous) solution $\boldsymbol{\chi}^*_B$ is found, this is fed to robot A, which from then on will solve a new optimization problem, which remains the same as the original

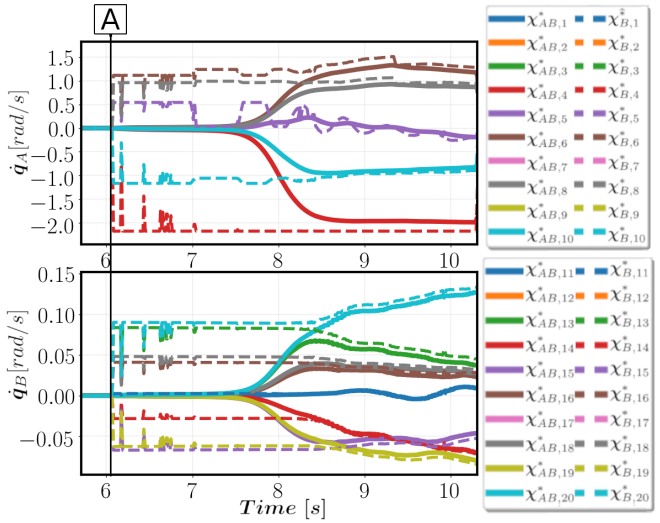

Fig. 6: Difference between $\boldsymbol{\chi}_B^*$ (dashed lines) and $\boldsymbol{\chi}_{AB}^*$ (solid lines) for robot A (top) and robot B (bottom), upon task activation (A), showing how the discontinuous variation is mitigated by $\boldsymbol{\chi}_{AB}^*$, which smoothly tends to $\boldsymbol{\chi}_B^*$.

one, but with an added last priority level in the SoT as:

$$\min_{\boldsymbol{\chi}} \|\boldsymbol{\chi} - \boldsymbol{\chi}_B^*\|^2 \tag{33}$$

which now allows to drive the solution from $\boldsymbol{\chi}_B^*$ to $\boldsymbol{\chi}_{AB}^*$. Indeed, the control sequences are always provided to both robots from robot A, through the solution of SoT A. By solving this problem, it is possible to smoothen the discontinuities between $\boldsymbol{\chi}_B^*$ and $\boldsymbol{\chi}_A^*$, while injecting the new SoT. As shown in Fig.4, we define $\boldsymbol{\chi}_{AB}^*$ as the optimal solution obtained from robot A, driving towards $\boldsymbol{\chi}_B^*$.

By keeping the highest priority of the SoT A unaltered (CLIK tasks) and acting on the constraints at velocity $\dot{\boldsymbol{q}}$ and acceleration levels $\ddot{\boldsymbol{q}}$, it is possible to achieve a smooth variation, by setting for robot A:

$$\alpha\boldsymbol{\chi}_{min}^s+(1-\alpha)\boldsymbol{\chi}_{min}^{t-\Delta t} \leq \boldsymbol{\chi} \leq \alpha\boldsymbol{\chi}_{max}^s+(1-\alpha)\boldsymbol{\chi}_{max}^{t-\Delta t} \tag{34}$$

where $\alpha \in \mathbb{R}$ is a time-dependent smoothly changing function from 0 to 1, and $\boldsymbol{\chi}_{min}^s, \boldsymbol{\chi}_{max}^s \in \mathbb{R}^n$ are the new limits triggered after the transition (see Fig. 1). For SoT B the constraints are instead directly changed to the new ones:

$$\boldsymbol{\chi}_{min}^s \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_{max}^s. \tag{35}$$

Therefore, the injection of $\boldsymbol{\chi}_B^*$ in SoT A enables the transition between two completely different problems (both in terms of tasks and constraints, as in Fig. 1), without requiring multiple swaps for each task modification as in [21], avoiding large computational increase.

E. Distributed Smooth Task Transitioning

Based on the tasks defined in Sec. III-B, we report here the HQP problems employed to test the transitions' smoothness. Robot's A original problem (before transitions) is:

$$\begin{aligned} 1 \quad & \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}^{CLIK}\boldsymbol{\chi} - \boldsymbol{b}^{CLIK}\right\|^2 \\ 2 \quad & \min_{\boldsymbol{\chi}} \|\boldsymbol{\chi}\|^2 \\ & s.t. \quad \boldsymbol{\chi}_{min} \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_{max} \end{aligned} \tag{36}$$

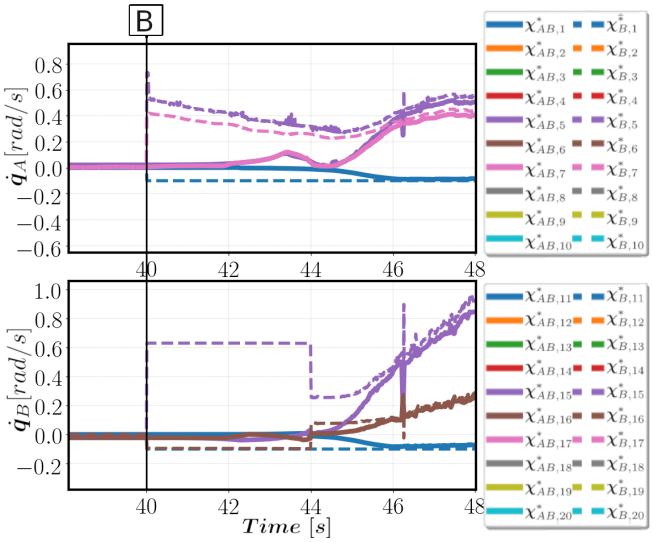

Fig. 7: Difference between $\boldsymbol{\chi}_B^*$ (dashed lines) and $\boldsymbol{\chi}_{AB}^*$ (solid lines) for robot A (top) and robot B (bottom), upon task activation (B), showing how the discontinuous variation is mitigated by $\boldsymbol{\chi}_{AB}^*$, which smoothly tends to $\boldsymbol{\chi}_B^*$.

whereas after the transition is triggered, **SoT A**:

$$\begin{aligned} 1 \quad & \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}^{CLIK}\boldsymbol{\chi} - \boldsymbol{b}^{CLIK}\right\|^2 \\ 2 \quad & \min_{\boldsymbol{\chi}} \|\boldsymbol{\chi} - \boldsymbol{\chi}_B^*\|^2 \\ & s.t. \quad \alpha\, \boldsymbol{\chi}_{min}^s \leq \boldsymbol{\chi} \leq \alpha\, \boldsymbol{\chi}_{max}^s \end{aligned} \tag{37}$$

On robot B instead, the first transition triggers the mobile bases problem **SoT B1**:

$$\begin{aligned} 1 \quad & \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}^{CLIK}\boldsymbol{\chi} - \boldsymbol{b}^{CLIK}\right\|^2 \\ 2 \quad & \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}_A^{base}\boldsymbol{\chi} - \boldsymbol{b}_A^{base}\right\|^2 + \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}_B^{base}\boldsymbol{\chi} - \boldsymbol{b}_B^{base}\right\|^2 \\ & s.t. \quad \boldsymbol{\chi}_{min}^s \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_{max}^s \end{aligned} \tag{38}$$

the second transition optimises arm postures **SoT B2**:

$$\begin{aligned} 1 \quad & \min_{\boldsymbol{\chi}} \left\|\boldsymbol{A}^{CLIK}\boldsymbol{\chi} - \boldsymbol{b}^{CLIK}\right\|^2 \\ 2 \quad & \min_{\boldsymbol{\chi}} \|\boldsymbol{A}_A^{pos}\boldsymbol{\chi} - \boldsymbol{b}_A^{pos}\|^2 + \min_{\boldsymbol{\chi}} \|\boldsymbol{A}_B^{pos}\boldsymbol{\chi} - \boldsymbol{b}_B^{pos}\|^2 \\ & s.t. \quad \boldsymbol{\chi}_{min}^s \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_{max}^s \end{aligned} \tag{39}$$

and the last one triggers **SoT B3** which just performs trajectory tracking, hence coincides with (36).

F. Joint Impedance Control

A lower-level impedance controller is added to provide an added layer of compliance at each joint, increasing safety in potential human-robot and robot-robot interaction:

$$\boldsymbol{\tau} = \boldsymbol{K}_{\boldsymbol{q_d}}(\dot{\boldsymbol{q}}^* - \dot{\boldsymbol{q}}_a) + \boldsymbol{K}_{\boldsymbol{q_p}}(\boldsymbol{q}^* - \boldsymbol{q}_a) + \boldsymbol{g}, \tag{40}$$

where $\boldsymbol{q}_a, \dot{\boldsymbol{q}}_a \in \mathbb{R}^n$ are the actual joint positions and velocities respectively, $\dot{\boldsymbol{q}}^*, \boldsymbol{q}^* \in \mathbb{R}^n$ are the desired optimal joint velocities and positions obtained from the HQP. Instead, $\boldsymbol{K}_{\boldsymbol{q_p}} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{K}_{\boldsymbol{q_d}} \in \mathbb{R}^{n\times n}$ identify the positive definite joint stiffness and damping matrices respectively, while $\boldsymbol{g} \in \mathbb{R}^n$ are the torques necessary for gravity compensation.

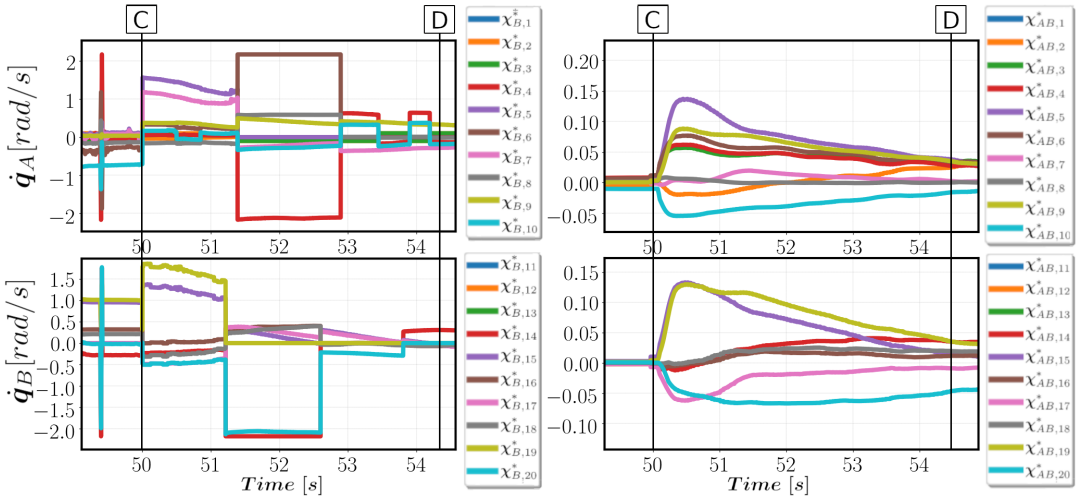

Fig. 8: Difference between χ_B^* (left-hand side) and χ_{AB}^* (right-hand side) for robot A (top) and robot B (bottom), upon task activation (C), showing how the discontinuous variation is mitigated by χ_{AB}^*, which smoothly tends to χ_B^*.

IV. Experiments

The proposed framework has been validated using two MObile Collaborative robotic Assistant (MOCA), which are mobile manipulators composed by a Robotnik SUMMIT-XL STEEL mobile platform, on top of which a Franka Emika Panda robotic arm is mounted. The overall platform consists of $n = 10$ DoF, and the end-effectors (EEs) used are a Panda Gripper and a PISA/IIT SoftHand. The manipulators are torque controlled and the actuation torques are obtained from the lower level JICs in (40). On the other hand, the mobile platforms are velocity controlled, hence the optimal velocities $\dot{\boldsymbol{x}}_{\boldsymbol{base}}$ are directly sent from the HQP.

The whole task sequence employed to test the transitions' smoothness is outlined in Fig. 5 and visible in the video[1]. The snapshots are taken close to steady-state conditions, when both robots accomplished their respective task. In particular, three transitions are tested with the proposed strategy and compared with the case in which the tasks are changed instantaneously. The first two plots in Fig. 5 show both EE's actual $\boldsymbol{x}_a$ against desired $\boldsymbol{x}^d$ trajectories, while the last two plots show the position and orientation of each mobile base frame. At time instant (A), the first transition is triggered, and *SoT B1* is the new hierarchy that aims at driving the position of the mobile bases towards $\boldsymbol{x}_{base,A}^d = \{0, -0.5, 0\}\ m$ and $\boldsymbol{x}_{base,B}^d = \{2, 0.5, 0\}\ m$, while maintaining higher EE's accuracy. At steady-state, the EE errors are lower than $10mm$, while the base errors are lower than $50mm$. Fig. 6 compares the difference between χ_B^* and χ_{AB}^* upon task activation, showing how the instantaneous and discontinuous variation is mitigated by χ_{AB}^*, which smoothly tends to χ_B^* as desired. At (B) the second transition is triggered and *SoT B2* is activated. Now the postural task is activated with $\boldsymbol{q}_{arm,A}^d = \boldsymbol{q}_{arm,B}^d = (\boldsymbol{q}_{max} - \boldsymbol{q}_{min})/2$ to minimize the displacements from actuators' mid-range. Fig. 7 compares again the sharp variation of χ_B^* and the smooth behaviour of χ_{AB}^* (here reported only for the most activated joints, for a clearer visualization). Finally, the last transition is triggered (C) and the new SoT drives both robots to the same location. Here, a discontinuity is introduced both in terms of change in SoT and in the step variation of the reference poses (Fig. 5). The final EE tracking error is below $2mm$. Fig. 8 shows on the left-hand side the solutions χ_B^*, while on the right-hand side χ_{AB}^*. It is clear how *SoT As* is able to smoothen the final solution, thanks to the insertion of (33) and to the possibility of performing constraints softening (34) on the double HQP problem (still solved as one-per-robot). Computational costs are compared in Fig. 9 (left), where Δt_A and Δt_B are the time intervals for the computation of χ_{AB}^* on robot A and χ_B^* on robot B, respectively, compared against Δt_C in which both χ_B^* and χ_{AB}^* are solved on the same machine, where indeed larger times are necessary. Fig. 9 (right), shows instead the shape of the chosen α function used in (34).

[1]https://youtu.be/ceM-HqMbgss?si=drOpsJOsrvSo40jq

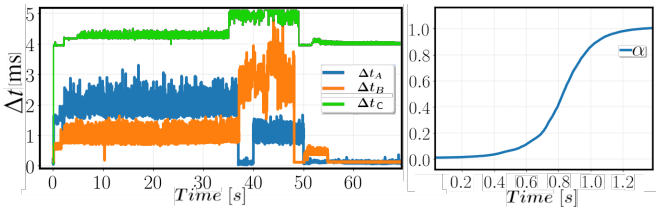

Fig. 9: Left: time intervals. Δt_A and Δt_B are the computational times of robot A and B respectively, while Δt_C is the case in which both χ_B^* and χ_{AB}^* are solved on the same machine. Right: α (34).

V. Conclusions

The proposed distributed processing approach was formulated to achieve smooth task transitions while always maintaining optimal and feasible solutions via the distributed problem decomposition and the continuous constraints modulation. Multiple task transitions have been evaluated in the experiments, which confirmed the smooth transitioning behavior between hierarchical SoT and constraints. Future studies will increase the number of robots, to assess the scalability and possible policies to decentralize the proposed framework on small subsets of larger multi-robot systems.

REFERENCES

[1] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.-G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino *et al.*, "Walk-man: A high-performance humanoid platform for realistic environments," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.

[2] M. Wang, W. Li, J. Luo, and U. Walter, "Coordinated hierarchical control of space robotic safe manipulation with load sharing," *Acta Astronautica*, vol. 202, pp. 360–372, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522005811

[3] L. Clemente, O. Villarreal, A. Bratta, M. Focchi, V. Barasuol, G. G. Muscolo, and C. Semini, "Foothold evaluation criterion for dynamic transition feasibility for quadruped robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4679–4685.

[4] F. Tassi, E. De Momi, and A. Ajoudani, "Augmented hierarchical quadratic programming for adaptive compliance robot control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3568–3574.

[5] P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, and A. Ollero, "A framework for set-based kinematic control of multi-robot systems," *Control Engineering Practice*, vol. 106, p. 104669, 2021.

[6] F. Tassi, E. De Momi, and A. Ajoudani, "An adaptive compliance hierarchical quadratic programming controller for ergonomic human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102381, 2022.

[7] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.

[8] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[9] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1283–1290.

[10] Y. Abe, M. Da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 249–258.

[11] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[12] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[13] F. Tassi, F. Iodice, E. De Momi, and A. Ajoudani, "Sociable and ergonomic human-robot collaboration through action recognition and augmented hierarchical quadratic programming," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 10 712–10 719.

[14] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, 2012.

[15] 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.

[16] F. Keith, P.-B. Wieber, N. Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3887–3892.

[17] J. Salini, S. Barthélemy, and P. Bidaud, "Lqp-based controller design for humanoid whole-body motion," in *Advances in Robot Kinematics: Motion in Man and Machine*, J. Lenarcic and M. M. Stanisic, Eds. Dordrecht: Springer Netherlands, 2010, pp. 177–184.

[18] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, 2012.

[19] G. Jarquín, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega, "Real-time smooth task transitions for hierarchical inverse kinematics," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 528–533.

[20] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Autonomous Robots*, vol. 40, pp. 17–31, 2016.

[21] S. Kim, K. Jang, S. Park, Y. Lee, S. Y. Lee, and J. Park, "Continuous task transition approach for robot controller based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1603–1610, 2019.

[22] G. Han, J. Wang, X. Ju, and M. Zhao, "Recursive hierarchical projection for whole-body control with task priority transition," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 312–11 319.

[23] 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.