

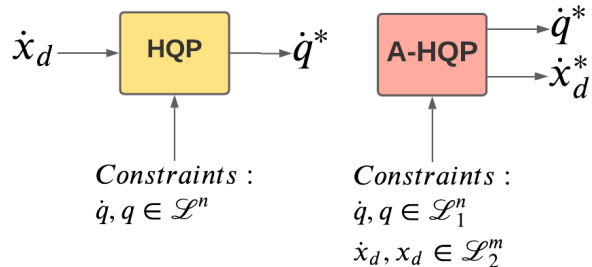
Augmented Hierarchical Quadratic Programming for Adaptive Compliance Robot Control

Francesco Tassi^{1,2}, Elena De Momi², and Arash Ajoudani¹

Abstract—Today’s robots are expected to fulfill different requirements originated from executing complex tasks in uncertain environments, often in collaboration with humans. To deal with this type of multi-objective control problem, hierarchical least-square optimization techniques are often employed, defining multiple tasks as objective functions, listed in hierarchical manner. The solution to the Inverse Kinematics problem requires to plan and constantly update the Cartesian trajectories. However, we propose an extension to the classical Hierarchical Quadratic Programming formulation, that allows to optimally generate these trajectories at control level. This is achieved by augmenting the optimization variable, to include the Cartesian reference and allow for the formulation of an adaptive compliance controller, which retains an impedance-like behaviour under external disturbances, while switching to an admittance-like behavior when collaborating with a human. The effectiveness of this approach is tested using a 7-DoF Franka Emika Panda manipulator in three different collaborative scenarios.

I. INTRODUCTION

Application domains of robotic systems are nowadays growing in complexity, imposing multiple requirements on the execution of various tasks simultaneously. To this purpose, redundant robots are finding always more use in common applications due to the possibility to achieve secondary objectives, or to acquire particular behaviours [1]–[3]. When dealing with multiple tasks however, it is possible that not all the objectives can be satisfied at the same time, leading to the necessity of defining different levels of priority, to allow controlling the robot under all the imposed constraints, which may depend on both its physical limitations (actuator capabilities, mechanical limits, etc.) and external environmental difficulties (obstacles, human behaviour and intentions, etc.). Plenty of works can be found in literature involving the management of multiple tasks using hierarchical control frameworks [3]–[5]. They are generally divided according to the methodology with which the task hierarchy is defined. A *non-strict* hierarchy [6], [7] is usually simpler to implement, since it often involves the weighting of different objectives, resulting however in much lower performances due to the inability of restricting a task at lower priority in the null-space of the tasks at higher priority. This is inadequate especially when the degree of redundancy increases, since a proper trade-off cannot be found. *Strict hierarchies* [3],



(a) Standard HQP based control algorithm for the Inverse Kinematics problem. (b) Proposed HQP based control algorithm that augments the output variable to include Cartesian trajectories.

Fig. 1: Block diagrams outlining inputs and outputs required for each version of the control algorithm used to solve the IK problem.

[4], [8]–[10] instead guarantee the fulfillment of the task having higher priority, by projecting the secondary task into the null-space of the primary one. Hierarchical Quadratic Programming (HQP) is the most common technique adopted to deal with *strict hierarchies* [11], [12], which translates the hierarchical control problem into the solution of a sequence of Quadratic Programming (QP) problems, without increasing the minimum solution obtained from the preceding higher priority tasks. Further works also consider dealing with both strict and non-strict hierarchies as in [13] where a Generalized Hierarchical Control framework defines the degree with which a lower-priority task is projected into the null-space of the higher priority task. Additionally, dynamically modifying the stack of tasks online during the execution can be of relevant importance in some applications [14]–[16] for achieving smooth and continuous transitions when inserting, removing, or swapping tasks. The following work is based on [17] in dealing with inequality constraints at any hierarchical level, which are used to express e.g. joint limits, obstacle avoidance, singularity avoidance, etc.

Classical HQP based Inverse Kinematics (IK) problems rely on reference trajectories defined at planning stage, that require updating for each environmental variation. In this work, the aim is to reduce this strict dependency from a precise plan, which can restrict the autonomy of the robot. The following approach augments the state of the least-square problem in order to include the Cartesian reference trajectory that will be obtained as an output, and that will reflect all those properties and policies specified through objectives and constraints in the hierarchy [18]. Thereby, the main features that can be achieved with this formulation are:

¹HRI² Lab, Dept. of Advanced Robotics, Istituto Italiano di Tecnologia, 16163, Genova, Italy. francesco.tassi@iit.it

²Department of Electronics Information and Bioengineering, Politecnico di Milano, Milan, Italy.

This work was supported by the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 871237 (SOPHIA).

- Optimal Cartesian reference trajectory generation at control level, which becomes useful with quick and repetitive tasks that need to adapt online with respect to e.g. human position, obstacles, etc. In this case, it is possible to avoid having to compute a new optimal trajectory at planning level for each modification in the environment, generating internally a trajectory that complies with the desired behaviour defined in the hierarchy of tasks.
- Creation of a human-robot collaborative framework in which the robot adapts to human forces by switching between impedance and admittance behaviour, through the variation of the desired Cartesian trajectories.

We start in Section II by recalling the classical HQP formulation for the problem resolution. Section III describes the control framework used to actuate the outputs of such problem, while the proposed control structure is presented in Section IV. Finally, Section V reports the experiments performed to evaluate this strategy and their results.

II. HIERARCHICAL QUADRATIC PROGRAMMING

The aim of this section is to briefly review the basic and most common formulation of the HQP scheme used for IK resolution. Let us first consider the IK problem of an n degrees of freedom robot, with desired joint velocity $\dot{\mathbf{q}} \in \mathbb{R}^n$, and task space velocity $\dot{\mathbf{x}} \in \mathbb{R}^m$

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^+ \dot{\mathbf{x}} \quad (1)$$

where $\mathbf{J}(\mathbf{q})^+ \in \mathbb{R}^{n \times m}$ is the pseudo-inverse of the task Jacobian matrix. Being mainly interested in the case in which the manipulator is redundant with respect to the defined task, the inequality $m < n$ is considered throughout the remainder of the work. In the redundant case, equation (1) actually represents one of the solutions of the following problem:

$$\min_{\dot{\mathbf{q}}} \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|^2. \quad (2)$$

The robot redundancy allows to define a secondary task that can be executed without affecting the performance of the primary one, which leads to the possibility of generating a set of tasks that can be executed hierarchically, exploiting the whole-body motion of the robot [19]. Indeed, by considering $k \in \{1, \dots, p\}$ levels of priority where the importance decreases with k down to the last task p , we ensure that the solutions found at level k are always strictly enforced at lower priority levels, which constitutes the main reason why we choose a *strict priority* scheme [17].

Another common approach is to adopt *soft priorities*, a simple solution that weights each task as in [6] and that does not require the solution of nested QP problems. Alternatively, weighting strategies can be used together with a strict formulation to define the importance between tasks at the same level, so that all the solutions are influenced by each other proportionally to their weights.

The generic and robot-independent HQP problem can be defined for the first hierarchical task as:

$$\begin{aligned} \min_{\chi} \frac{1}{2} \|\mathbf{A}_1 \chi - \mathbf{b}_1\|^2 \\ \text{s.t. } \mathbf{C}_1 \chi \leq \mathbf{d}_1 \\ \mathbf{E}_1 \chi = \mathbf{f}_1 \end{aligned} \quad (3)$$

with generic matrices $\mathbf{A}_k, \mathbf{C}_k, \mathbf{E}_k \in \mathbb{R}^{s \times s}$, and vectors $\mathbf{b}_k, \mathbf{d}_k, \mathbf{f}_k \in \mathbb{R}^s$, while $\chi \in \mathbb{R}^s$ is the generic variable to optimize, subject to equality and inequality constraints. The solution is χ_1^* , which is then used to constrain the successive QP iterations. Indeed, the generic k^{th} task can be written as:

$$\begin{aligned} \min_{\chi} \frac{1}{2} \|\mathbf{A}_k \chi - \mathbf{b}_k\|^2 \\ \text{s.t. } \mathbf{C}_1 \chi \leq \mathbf{d}_1, \dots, \mathbf{C}_k \chi \leq \mathbf{d}_k \\ \mathbf{E}_1 \chi = \mathbf{f}_1, \dots, \mathbf{E}_k \chi = \mathbf{f}_k \end{aligned} \quad (4)$$

where the previous $1, \dots, k-1$ solutions are considered through the optimality conditions between successive tasks $\mathbf{A}_{k-1} \chi = \mathbf{A}_{k-1} \chi_{k-1}^*$, whose demonstration is provided in [17]. In this way, the optimality of the tasks with higher priority is not altered by the actual solution, and it can be added in (4) as a set of equality constraints by considering $\mathbf{E}_1 = \mathbf{0}, \mathbf{f}_1 = \mathbf{0}$, up to $\mathbf{E}_k = \mathbf{A}_{k-1}, \mathbf{f}_k = \mathbf{A}_{k-1} \chi_{k-1}^*$.

III. CONTROL SCHEME

Aiming at an improved human-robot collaborative framework, we chose a lower level *joint impedance controller*, providing the actuator torques as:

$$\boldsymbol{\tau} = \mathbf{K}_{q_d}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_{q_p}(\mathbf{q}_d - \mathbf{q}) + \mathbf{g}(\mathbf{q}) \quad (5)$$

where $\mathbf{q}_d \in \mathbb{R}^n$ are the desired joint positions obtained by integrating the optimal desired joint velocity, $\mathbf{K}_{q_p} \in \mathbb{R}^{n \times n}$ and $\mathbf{K}_{q_d} \in \mathbb{R}^{n \times n}$ are the positive definite joint stiffness and damping matrices respectively, while $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity compensation torque vector. It must be noted however, that by implementing the impedance controller locally for each actuator, we can select \mathbf{K}_{q_p} and \mathbf{K}_{q_d} as diagonal matrices, but in this case it is not possible to obtain a precise Cartesian stiffness at the end-effector. To this purpose, it is possible to consider an optimization problem that minimizes the difference with respect to the desired Cartesian stiffness [20], [21], allowing to define the best approximation alternative. A Closed-loop IK (CLIK) scheme is used to recover from position errors between the desired and actual behaviour, rewriting equation (2) as:

$$\min_{\dot{\mathbf{q}}} \|\mathbf{J}\dot{\mathbf{q}} - (\dot{\mathbf{x}}_d + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}_a))\|^2 \quad (6)$$

where $\dot{\mathbf{x}}_d, \mathbf{x}_d \in \mathbb{R}^m$ are respectively the desired Cartesian velocities and positions, $\mathbf{x}_a \in \mathbb{R}^m$ is the actual Cartesian position and $\mathbf{K}_p \in \mathbb{R}^{m \times m}$ is the positive definite diagonal gain matrix that is responsible for the error convergence.

IV. AUGMENTED HQP FORMULATION

In this section, we describe the state augmentation process for the IK problem of a redundant robot through HQP, providing the reasons and purposes of this alternative framework, along with its possible advantages and limitations.

A. State Augmentation

The block scheme in Fig. 1a depicts the inputs and outputs of the algorithm described in Section II. The main input is the desired Cartesian trajectory to be followed by the robot, together with the constraints of the optimization variable, which is then obtained as the output. The proposed scheme instead can be identified by Fig. 1b, in which the desired trajectories are no longer provided, being instead part of the controller outcome. In this case, the only inputs are the constraints, which can now be defined also with respect to the Cartesian coordinates. The optimal output set of Cartesian trajectories $\dot{\mathbf{x}}_d^*$ will then follow the desired criteria defined in the hierarchy. One limitation is clearly related to constraints definition, which must reflect the non-emptiness and convexity properties of QP. Non-convex constraints due to obstacles for example, can be linearized to approximate the workspace, while imposing a desired behaviour in the hierarchy (e.g. reach a target avoiding forbidden regions).

The idea here is to increase the robot's adaptability by defining its desired behaviour instead of relying on a pre-planned trajectory. This behaviour is defined through an hierarchical stack of tasks, allowing for the desired Cartesian trajectory to be generated optimally. Fig. 1b shows this concept, obtained by augmenting the state variable $\chi \in \mathbb{R}^{n+m}$:

$$\chi = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{x}}_d \end{bmatrix}. \quad (7)$$

In this way, through the constraints it is possible to define the feasible region in which the robot must operate, and through the objective functions organized in hierarchical manner it is possible to impose a desired behaviour. To this purpose, it is useful to consider a dynamic stack of tasks, that can be swapped or modified online during the execution, based on the interaction with the environment and the user, as it will be further discussed in the experiments in Section V. The augmentation of the kinematic task leads to rewriting the problem in (6), which considering the reference Cartesian positions as obtained by numerical integration of the reference Cartesian velocities as:

$$\mathbf{x}_d = \int_{t_0}^t \dot{\mathbf{x}}_d dt \simeq \mathbf{x}_d(t - \Delta t) + \dot{\mathbf{x}}_d(t)\Delta t, \quad (8)$$

then becomes:

$$\min_{\dot{\mathbf{q}}, \dot{\mathbf{x}}_d} \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d - \mathbf{K}_p(\mathbf{x}_d(t - \Delta t) + \dot{\mathbf{x}}_d\Delta t - \mathbf{x}_a)\|^2 \quad (9)$$

which can be rewritten in augmented form as:

$$\begin{aligned} \min_{\dot{\mathbf{q}}, \dot{\mathbf{x}}_d} \left\| \begin{bmatrix} \mathbf{J} & -(\mathbf{I} + \mathbf{K}_p\Delta t) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{x}}_d \end{bmatrix} - \mathbf{K}_p(\mathbf{x}_d(t - \Delta t) - \mathbf{x}_a) \right\|^2 \\ = \min_{\chi \in \Omega} \|\mathbf{A}_1\chi - \mathbf{b}_1\|^2 \end{aligned} \quad (10)$$

where $\Omega \subset \mathbb{R}^{n+m}$ is a non-empty convex set. Being the IK now written in QP form as (3), it can be included in the hierarchy.

B. Ill-conditioning issues

This version shares the same ill-conditioning issues with the classical HQP described in Section II. Indeed, when a task becomes unfeasible with respect to the ones holding a higher priority, it might happen that the pseudo-inverse solutions of (1) grow unbounded due to some ill-conditioning problems [22]. This issue is faced in [17] by using a weighted regularization term that is important for numerical stability:

$$\min_{\chi} \frac{1}{2} \|\mathbf{A}_1\chi - \mathbf{b}_1\|^2 + \frac{1}{2} \rho_1^2 \|\chi\|^2, \quad \rho_1 \in \mathbb{R}^*. \quad (11)$$

In our case, it is possible to define a new task responsible only for the regularization of one or both the elements of the augmented vector (7), by defining it in QP form similarly to how it was done in (10). This also improves the smoothness of the output variables while transitioning between tasks.

C. Objective functions for task definition

Having augmented the state variable, it is possible to define new sets of objectives and constraints that now depend on both the Cartesian coordinate of the end-effector (or any other point along the kinematic chain) and the joint coordinates. When it is necessary to reach a target pose for example, since \mathbf{x}_d is now an output of the augmented HQP, meaning we are only in charge of its bounds through constraints definition, we can impose a target pose $\mathbf{x}_t \in \mathbb{R}^m$ by setting the desired behaviour of \mathbf{x}_d through:

$$\begin{aligned} \min_{\chi} \|\mathbf{x}_d - \mathbf{x}_t\|^2 &= \min_{\chi} \|\mathbf{x}_d(t - \Delta t) + \dot{\mathbf{x}}_d\Delta t - \mathbf{x}_t\|^2 = \\ \min_{\chi} \left\| \begin{bmatrix} \mathbf{0}_{m \times n} & \mathbf{I}_{m \times m}\Delta t \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{x}}_d \end{bmatrix} - (\mathbf{x}_t - \mathbf{x}_d(t - \Delta t)) \right\|^2. \end{aligned} \quad (12)$$

In this way, \mathbf{x}_t will dictate the planning of \mathbf{x}_d , which consequently affects \mathbf{x}_a through the CLIK (10). Similarly, the definition of a postural task is also often necessary, exploiting the redundancy of the robot to keep the joint space configuration as similar as possible to the desired one, and it is defined analogously to (12) from

$$\min_{\chi} \|\mathbf{q}_{ref} - \mathbf{q}_a\|^2, \quad (13)$$

$\mathbf{q}_{ref}, \mathbf{q}_a \in \mathbb{R}^n$ being respectively the desired and actual joint positions. Many other classical behaviours can be formulated as optimization objectives, such as: minimum joint displacement, singularity avoidance, etc. In addition, with this augmented version it is easy to define other objectives directly as a function of Cartesian quantities, such as: maximum distance from obstacles, minimum Cartesian displacement, Cartesian acceleration/jerk reduction, etc.

D. Constraints

In addition to the already existing technological limits inherent to the machine, task feasibility is limited in the real scenario of a robot working in an environment with obstacles and interaction bounds. In this paragraph, only the most important constraints will be presented.

Kinematic constraints: these are generally represented by the actuators range of motion, velocity and acceleration constraints, which can be formulated through (8) as:

$$\begin{aligned} \mathbf{q}_{min} &\leq \mathbf{q}(t - \Delta t) + \dot{\mathbf{q}}(t)\Delta t \leq \mathbf{q}_{max} \\ \dot{\mathbf{q}}_{min} &\leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \\ \ddot{\mathbf{q}}_{min} &\leq \frac{\dot{\mathbf{q}}(t) - \dot{\mathbf{q}}(t - \Delta t)}{\Delta t} \leq \ddot{\mathbf{q}}_{max} \end{aligned} \quad (14)$$

$$\begin{aligned} \mathbf{x}_{dmin} &\leq \mathbf{x}_d(t - \Delta t) + \dot{\mathbf{x}}_d(t)\Delta t \leq \mathbf{x}_{dmax} \\ \dot{\mathbf{x}}_{dmin} &\leq \dot{\mathbf{x}}_d \leq \dot{\mathbf{x}}_{dmax} \\ \ddot{\mathbf{x}}_{dmin} &\leq \frac{\dot{\mathbf{x}}_d(t) - \dot{\mathbf{x}}_d(t - \Delta t)}{\Delta t} \leq \ddot{\mathbf{x}}_{dmax} \end{aligned} \quad (15)$$

and similarly to what was shown before, it is possible to describe all constraints as a function of χ in augmented form:

$$\frac{1}{\Delta t} \begin{pmatrix} \mathbf{q}_{min} - \mathbf{q}(t - \Delta t) \\ \mathbf{x}_{dmin} - \mathbf{x}_d(t - \Delta t) \end{pmatrix} \leq \chi \leq \frac{1}{\Delta t} \begin{pmatrix} \mathbf{q}_{max} - \mathbf{q}(t - \Delta t) \\ \mathbf{x}_{dmax} - \mathbf{x}_d(t - \Delta t) \end{pmatrix} \quad (16)$$

$$\chi_{min} \leq \chi \leq \chi_{max} \quad (17)$$

$$\begin{pmatrix} \dot{\mathbf{q}}(t - \Delta t) + \dot{\mathbf{q}}_{min}\Delta t \\ \dot{\mathbf{x}}_d(t - \Delta t) + \dot{\mathbf{x}}_{dmin}\Delta t \end{pmatrix} \leq \chi \leq \begin{pmatrix} \dot{\mathbf{q}}(t - \Delta t) + \dot{\mathbf{q}}_{max}\Delta t \\ \dot{\mathbf{x}}_d(t - \Delta t) + \dot{\mathbf{x}}_{dmax}\Delta t \end{pmatrix} \quad (18)$$

Dynamic constraints: on the other hand, dynamic constraints are related to the actuators torque limits imposed by the manufacturer $\tau_{min} \leq \tau \leq \tau_{max}$ and it can be possible to address for these elements first in terms of joint accelerations exploiting the inverse dynamics of the manipulator, and then with respect to joint velocities through numerical integration

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{m}(\mathbf{q}, \dot{\mathbf{q}}) \quad (19)$$

where $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive-definite inertia matrix, while $\mathbf{m}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is responsible for Coriolis, centrifugal and gravitational terms.

Interaction constraints: we only mention here the possibility of creating a kinematic loop, which allows to put in contact two points along the kinematic chain of the robot, resulting in the equality of their respective joint velocities [6]. Being \mathbf{J}_1 and \mathbf{J}_2 the Jacobian matrices of the two points:

$$(\mathbf{J}_1 - \mathbf{J}_2)\dot{\mathbf{q}} = 0 \quad (20)$$

V. EXPERIMENTS & RESULTS

The proposed framework was tested on the torque-controlled 7-DoF Franka Emika Panda manipulator equipped with a PISA/IIT SoftHand. The low-level torque control frequency of the manipulator is 1kHz, while the controller computer is an Intel Core i7 @ 1.80 GHz \times 8; 8 GB of RAM.

A. Experiment 1: Human-Robot Collaborative Framework

As anticipated, the aim is to identify a generic collaborative scenario in which the robot interacts with the human and responds accordingly, based on the inputs from the environment and the operator. The task considered is a common and generic industrial assembly task, in which the robot is used to: bring the pieces to the assembly workbench in order to start the assembly, help the operator during the most difficult and stressful tasks, respond compliantly with respect to the forces perceived. The overall workspace is

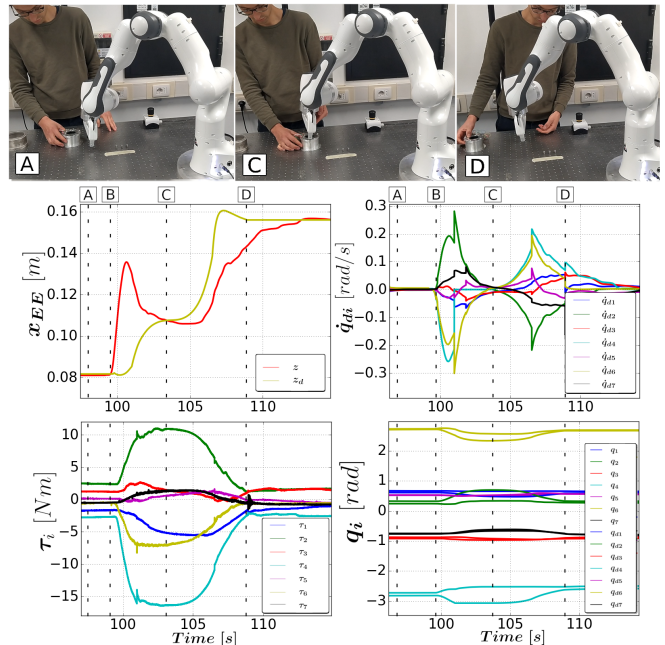


Fig. 2: *Experiment 1.* The human pulls the robot's EE upwards along the positive z -axis (B) and keeps a constant pressure on the robot until (C), when the actuator torques start diminishing and the human feels the effort reduction while keeping the same position of the EE. The plots show the actual z_a (red) vs desired z_d (yellow) EE Cartesian z -coordinate (top left), the actuation torques τ_i obtained from the joint level impedance control (5) (bottom left), the desired joint velocity $\dot{\mathbf{q}}_d$ obtained from the A-HQP (top right) and the actual \mathbf{q} vs desired \mathbf{q}_d joint positions (bottom right).

shown in Fig. 2 and it can be easily defined in terms of constraints of the Cartesian variable as in (16).

Firstly, the robot is required to bring a piece from its initial position to the actual shared workspace. The peculiarity lies in the fact that we now do not have an already-defined trajectory at motion planning level to reach the desired location. Indeed, we exploit the augmented formulation as shown in Fig. 1b in order to define only the workspace constraints, and to set a desired behaviour for the specific task. In this experiment, we define the objective of minimum distance with respect to the human position, which can be specified as in (12), allowing to reach the correct region of the workspace. Besides the IK constraint (10), we impose in the hierarchy the regularization term specified in Section IV-B useful for numerical stability and continuity of the solution. Similarly, it is possible to include obstacle avoidance features in the stack, in case obstacles are present. The constraints considered are (16), (17) and (18). Finally, the desired Cartesian trajectory \mathbf{x}_d is generated online as an output and is optimal with respect to the lower priority tasks, satisfying all the desired properties, provided that the degree of redundancy of the application is large enough. Once the robot has reached the operator, the second part of the experiment can start, which consists of the collaborative task itself. To this purpose, it is important to refer to what was mentioned in Section IV-A since it is useful to consider a dynamic stack of tasks, whose components can be swapped or modified online during

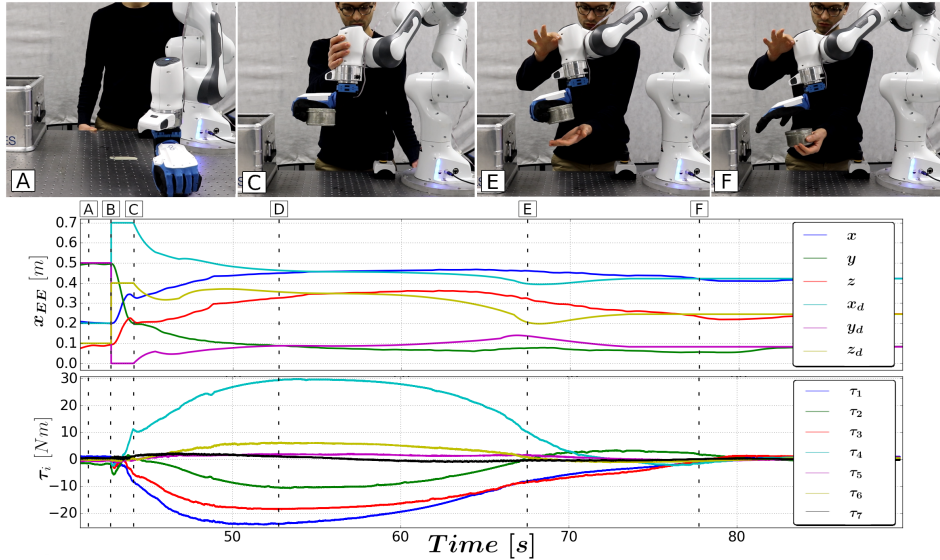


Fig. 3: *Experiment 2*. The human stops the end-effector before it has reached its target location, leading to an increase in actuation torques τ (bottom plot), that are then reduced by the proposed scheme. In particular, the desired \mathbf{x}_d and actual \mathbf{x} Cartesian trajectories are shown (top plot). After the new target reference \mathbf{x}_t is set (B) using (12), the human physically intervenes on \mathbf{x}_a in (6), triggering the *free* behaviour and the variation of \mathbf{x}_d (C), leading to velocity inversion and subsequent torques reduction (D) up to the new equilibrium (F).

the execution, based on the interaction with the environment and the user. In the present work, we proceed by considering two main robot behaviours. The first one, that we henceforth define as *constrained* behaviour, is the behaviour described up to now in the first step of the experiment, in which the robot is required to optimize its trajectory online in order to reach a point or a region, obtained as the minimum of a desired objective function, which then remains constant. The second instead, which we will refer to as *free* behaviour, allows for the variation of the desired Cartesian trajectory \mathbf{x}_d and omits any other objective that binds it, i.e. (12), leaving mainly the IK task (10). This allows to formulate an *adaptive compliance control* framework with respect to the forces perceived by the robot, so to respond with an impedance-like behaviour when the forces are coming from the environment as disturbances, while to achieve an admittance-like behaviour when the forces are continuously exerted by the operator. This is obtained by switching between the two behaviours, based on a threshold defined on the error between the actual and desired joint positions $e_q = \|\mathbf{q} - \mathbf{q}_d\|_\infty$ passed to the impedance controller in (5), so that when the operator is pushing steadily on some of the robot's joints, aiming at modifying the position of the robot, the *free* behaviour is activated. Therefore, the reference Cartesian trajectories start to change, providing an admittance-like response and thus accommodating for the operator's push. This is useful for example when the operator needs to move the end-effector to work on the piece, or in the case he needs to bring the robot to a more comfortable or ergonomic position. Indeed, it is realistic to think that the operator might need to move or adjust the robot during tasks or in-between operations, and this framework aims at increasing the autonomy of the robot with respect to the case of a fixed plan or trajectory.

For this reason, we consider the case in which the human is pulling the robot's end-effector upwards along the positive z -axis, with the intention of lifting the end-effector after a generic operation is complete (e.g. deburring, polishing, painting etc.), or just of increasing the comfort level during a collaborative assembly task. The details of the experiments are shown in Fig. 2. Initially the robot is at steady-state (A), but after being perturbed (B) the joint position error of Fig. 2 (bottom right) increases, overcoming the threshold and activating the *free* behaviour, which leads to the variation of \mathbf{x}_d , that tries to reduce the Cartesian error imposed in the CLIK scheme in (6) until crossing the actual values \mathbf{x}_a (C) and thus swapping the sign of $\dot{\mathbf{x}}_d$ in Fig. 2 (top right). This translates into the reduction of the actuators torques obtained with (5) shown in the bottom left plot of Fig. 2, which decrease down to zero while the operator constantly reduces the applied effort, until the robot switches back to the *constrained* behaviour where the new equilibrium point is now considered (D). In conclusion, thanks to the augmented state, and being the actual Cartesian pose \mathbf{x}_a in (6) now constrained by the human, we were able to exploit the variation of $\dot{\mathbf{x}}_d$ in the CLIK to obtain a variation in the joint velocities which counteracts the accumulated torques. As visible from Fig. 2 (top left), it was possible to accommodate for the human intention of moving upwards the end-effector of approximately 80mm. This was achieved using the same scheme of the first experimental stage, that is therefore capable of both rejecting sudden external perturbations, which do not lead to a sufficient increase in joint position error, while still complying with human actions.

B. Experiment 2: Robot adaptation to human actions

A second experiment is useful to better underline the potentials of the suggested framework in a collaborative sce-

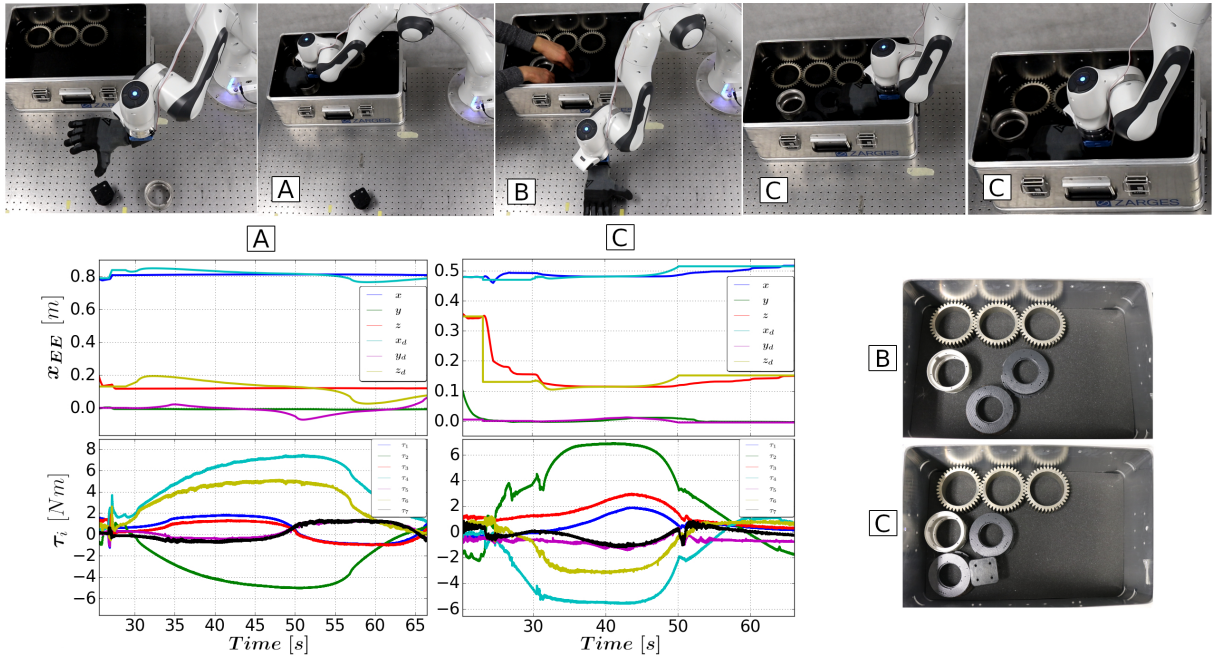


Fig. 4: *Experiment 3*. Pick & place of different items in a box (A, C), while compacting the ones placed by the human (B) using the proposed force adaptation scheme. The upper plots show the actual x_a vs desired x_d Cartesian trajectories, while the bottom plots show the actuation torques τ_i . On the right, the snapshots after the human placement (B) and the robot adjustments (C).

nario. Indeed, when performing e.g. a collaborative industrial assembly, the human timings for each sub-task completion can be variable or unpredictable, especially in applications relying on highly specialized labor. Hence, the operator should be able to intervene, correct or even anticipate the robot. We consider here the case in which the human might need to anticipate the robot before it has reached its final target pose. For example, if the robot is required to pick an object and pass it to the operator, but the operator's position has changed, he can intervene, causing to switch to the *free* behaviour after the applied force is felt. The chosen setup is shown in Fig. 3 where the pick and place task starts from an initial pose (A) and should reach the box on the left side of each photo. The plots show this task in detail, in which the robot end-effector starts at equilibrium (A) and is required from time instant (B) to reach the point $(0.7, 0.0, 0.4)$, keeping the same orientation. The robot approaches the target point until the operator intervenes in (C), opposing the movement and increasing the torques. The reference Cartesian variation is triggered, leading to \dot{x}_d eventually swapping sign (D) and reducing the accumulated torques felt by the human (E). Finally, the new equilibrium position reached $(0.42, 0.08, 0.24)$ (F) is a result of the operator intervention, which has reduced the end-effector overall covered path of more than 300 mm.

C. Experiment 3: Collaborative box filling

The same control framework is used here in another common industrial scenario, in which the robot and the human collaborate to compactly fill a box with different objects (Fig. 4). When the robot places each piece in the box in (A) and (C) it uses the same behaviour of *Experiment 2*

to push against the other elements, adapting the torques and thus compacting the elements so that the operator in (B) does not have to worry about precise positioning. Fig. 4 shows the detailed task, with the same plots seen before, allowing to compact the items placed in the box (B vs C).

VI. CONCLUSIONS AND DISCUSSIONS

In the context of HQP, we have proposed an alternative method to deal with the IK problem of a redundant robot. This framework allows to obtain the desired Cartesian trajectories as output of the control algorithm, exploiting the constraints and the objectives definition in order to achieve the desired robot behaviour. This can prove useful in those applications in which it is necessary to repetitively change the reference trajectory online due to environmental changes or obstacles. On the other hand, the proposed control also allows to obtain a flexible framework with respect to human actions and intentions, which can adapt the robot's compliant behaviour thanks to the reference variation. The main limitations of this framework are related to constraint definition, that must comply with QP feasibility properties. The constraints can be updated at each iteration, defining a new feasible region, but they must be written in linear form to be solved as in (4). Another issue of the augmentation process is the increased computational cost, that is however limited when increasing the state of dimension m , but still proportional to the number of constraints and objectives. Future developments will account for the human counterpart through ergonomic functions, that will allow to minimize physical stress and discomfort.

REFERENCES

- [1] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 842–868, 1977.
- [2] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.
- [3] B. Siciliano and J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pp. 1211–1216 vol.2, 1991.
- [4] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Rob. Res.*, vol. 6, no. 2, p. 3–15, Jul. 1987. [Online]. Available: <https://doi.org/10.1177/027836498700600201>
- [5] F. Flacco and A. De Luca, "Discrete-time redundancy resolution at the velocity level with acceleration/torque optimization properties," *Robotics and Autonomous Systems*, vol. 70, pp. 191 – 201, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889015000329>
- [6] 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.
- [7] Y. Abe, M. Silva, and J. Popovic, "Multiobjective control with frictional contacts," 01 2007, pp. 249–258.
- [8] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, vol. 2, 2004, pp. 764–780 Vol. 2.
- [9] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010. [Online]. Available: <https://doi.org/10.1145/1778765.1781157>
- [10] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, ser. Addison-Wesley series in electrical and computer engineering: Control engineering. Addison-Wesley Publishing Company, 1991. [Online]. Available: <https://books.google.it/books?id=hp4QAQAAMAAJ>
- [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] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Autonomous Robots*, vol. 40, no. 1, pp. 17–31, 2016.
- [14] 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)*. IEEE, 2013, pp. 528–533.
- [15] 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.
- [16] F. Keith, P.-B. Wieber, N. Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discrete task scheduling operations," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3887–3892.
- [17] O. Kanoun, F. Lamiroux, 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] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.
- [19] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "Opensot: a whole-body control library for the compliant humanoid robot coman," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6248–6253.
- [20] A. Albu-Schaffer, M. Fischer, G. Schreiber, F. Schoeppe, and G. Hirzinger, "Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 4. IEEE, 2004, pp. 3295–3301.
- [21] A. Rocchi, E. Hoffman, E. Farnioli, and N. Tsagarakis, "A whole-body stack-of-tasks compliant control for the humanoid robot coman," 2014.
- [22] A. Dietrich, C. Ott, and J. Park, "The hierarchical operational space formulation: Stability analysis for the regulation case," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1120–1127, 2018.