Explicit compliance and safety on torque controlled robots for physical interaction

Mathieu Celerier, Bastien Muraccioli, Mehdi Benallegue, Yue Hu, Rafael Cisneros Limón, Hiroshi Kaminaga, Gentiane Venture

Explicit Compliance and Safety on Torque Controlled Robots for Physical Interaction

Mathieu Celerier[1], Bastien Muraccioli[1], Mehdi Benallegue[1], Yue Hu[3], Rafael Cisneros Limón[1], Hiroshi Kaminaga[1], Gentiane Venture[1,2]

***Abstract*—Robots operating in close physical contact must reconcile high-precision motion with context-dependent compliance and uncompromising safety. We introduce a quadratic programming (QP) torque-control framework that incorporates external torque estimates into the inverse dynamics to compute accurate torque commands, enabling proper acceleration tracking under external disturbances. The proposed QP framework embeds an explicit compliance parameter, allowing selective and continuously tunable compliance and effective inertia shaping at both the task and joint levels.**

We further revisit the parametrization of acceleration-based formulations for position and velocity constraints, removing their dependency on the control frequency. Specifically, we propose a second-order velocity damper based on two intuitive parameters, and provide an experimental analysis of their effects. This study reveals that classical formulations relying on the control loop frequency are not suitable and provides guidance on selecting appropriate parameters for a given system.

The framework is validated on a Kinova Gen3 across diverse interaction scenarios, ranging from stiff, null-space compliance to full-body compliance, demonstrating advanced behaviors such as task-space inertia shaping, while ensuring safety under unpredictable external forces.

These results suggest that coupling explicit compliance with external force-aware QP optimization offers a practical and versatile approach to achieving precise and adaptive physical human–robot interaction.

***Index Terms*—Physical Human-Robot Interaction, Robot Safety, Sensor Fusion**

I. Introduction

INDUSTRIAL manipulators have traditionally operated in caged workcells, executing pre-planned trajectories from a set of handcrafted waypoints or virtual barrier defined by the operator, and executing it with high-gain position or velocity control to guarantee precision and repeatability [1], to avoid unintended collisions or interaction with their environment and each other [2]. Such stiff control loops generate large corrective torques to compensate for deviations from the reference joint trajectory, and therefore can only be used in situation with no unplanned contact. However, with Industry 5.0 emphasizing close physical collaboration between humans and robots, robots migrating to warehouses, hospitals, and homes, frequent contact with uncertain, human-populated surroundings becomes inevitable. In these settings, the very speed and power that ensure precision can cause severe, even fatal, accidents during unexpected collisions. A new generation of collaborative robots therefore demands control strategies that reconcile high performance with uncompromising safety.

The first safety principle is to limit the robot's ability to generate harmful forces [3] such as applying large torque or moving aggressively, yet many tasks still require millimetric accuracy and repeatability. Hence, the central dilemma: the robot must be both *precise* and *compliant*.

- **Precision**: Strict adherence to kinematic constraints (self-collision, joint and Cartesian limits) and be able to precisely follow planned trajectories that require reliability and repeatability.
- **Compliance**: Deliberate reduction of internal/external forces to prevent harmful forces, allowing gentle deviations from, or adaptation of planned motions under external disturbances without compromising safety.

Typical impedance/admittance control [4] achieves low contact forces but may produce large tracking errors or leads the robot to drift in unsafe configuration. Conversely, high-gain position control ensures accurate motions but can cause dangerous impacts due to the excessive generated forces. Balancing these antagonistic objectives remains a defining challenge in physical human–robot interaction (pHRI) [5].

An *ideal* controller able to balance these conflicts enables the robot to remain compliant by default and *increase its stiffness only when a constraint approaches its limit*. This *selective stiffness* maximizes safety by limiting interaction forces until a constraint breach is imminent, while allowing for preserved accuracy when it truly matters for a task.

It's important to note that throughout this paper, we use the terms *stiffness* and *compliance* in a broader sense. In our context, these terms denote the robot's physical response to external wrenches. Besides the classical stiffness stemming from elastic systems or high feedback gains that resists deviations, we include the concept of *effective inertia* emulation, characterizing how external wrenches produces accelerations: a system with large apparent inertia reacts little to a given force (stiff), whereas low inertia produces large accelerations (compliant). The proposed framework modulates both aspect transparently.

All supplemental materials can be found compiled at **https://youtu.be/T1H7nLsjT4Y**

[1]M. Celerier, B. Muraccioli, M. Benallegue, R. Cisneros L., H. Kaminaga and G. Venture are with CNRS-AIST JRL (Joint Robotics Laboratory), IRL3218, Department of Information Technology and Human factors, National Institute of Advanced Industrial Science and Technology (AIST), AIST Tsukuba Headquarters and Information Technology Collaborative Research Center, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8560, Japan, {mathieu.celerier, bastien.muraccioli, mehdi.benallegue, rafael.cisneros, hiroshi.kaminaga}@aist.go.jp

[2]G. Venture is also with the Department of Mechanical Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan, venture@g.ecc.u-tokyo.ac.jp

[3]Y. Hu is with the Department of Mechanical & Mechatronics Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada, yue.hu@uwaterloo.ca

Finally, such framework should remain flexible in defining which tasks can be compliant and which require higher stiffness. One prominent example is when the primary task does not use all degrees of freedom, remaining (null-space) directions can stay compliant, allowing high stiffness on critical tasks (e.g., precise or force-sensitive operations), while keeping compliance where it does not compromise task performance or safety. Existing implementations, either force-sensor based [6] or purely model-based [7], lack a unified mechanism to *adapt* stiffness in real time while guaranteeing safety.

Modern robot control software is often organized into layers, with each layer building upon the safety and performance guarantees of the layers below [8].

Highest level: This layer encompasses semantics, mission planning, and behaviour generation. Safety mechanisms here operate at the planning stage: they implement high-level collision avoidance and ensure that tasks and actions adhere to operational rules [9], [10].

Mid-level: This layer generates the reference trajectories for the lowest layer. It coordinates tasks, ensures stability, and enforces safety in real time through strict kinematic, dynamic, and collision-avoidance constraints, continuously monitoring the robot state [11], [12].

Lowest level: This layer deals with joint-level control (current, torque, position). It guarantees stability and bandwidth. Classical impedance, admittance, or PID schemes excel either at force regulation or at trajectory tracking, but none alone can realise selective stiffness with safety guarantees.

In practice, the low-level controller must:

1) Respect the kinematic and safety limits.

2) Achieve the required precision and reliability imposed by the mid-level, provided this does not interfere with the first objective.

3) Limit both internally generated and externally applied forces [13], provided this does not interfere with the first two objectives.

However, the low-level control loop is aware only of the reference trajectories and has no direct knowledge of which constraints are close to being violated. As a result, to guarantee safety on its own, the low-level must default to a stiff behaviour at all times. This prevents unintended limit violations but also removes the possibility of selective compliance. Deciding when to adjust stiffness or compliance must therefore occur at least at the mid-level, which can issue references that explicitly encode compliance as a desired task dynamic.

A further difficulty is that activating or deactivating constraints often introduces non-smooth dynamics. Sensor noise can cause chattering when the robot state is close to a limit [14]. This creates the need for a robust control solution that can handle abrupt stiffness changes while accurately tracking mid-level references that mix safe, compliant, and precise behaviour.

A. Related Works

Passive elements, back-drivable gear trains, elastic joints, and soft materials, introduced to balance safety and performance, absorb shocks and limit contact forces without sensing [15]. While it allows for natural compliance, these systems cannot display high-stiffness therefore loosing performance on precision and repeatability, essential for many applications. Thus, most collaborative manipulators (e.g., *Panda*, *Gen3*, *LBR iiwa*) retain only modest intrinsic flexibility and instead rely on high-bandwidth torque sensing and/or direct torque control for finer interaction control [16], [17], [18].

Force-feedback schemes, such as impedance, admittance, and their dynamical-systems variant [19], shape the closed-loop behaviour to maintain forces below task-specific thresholds either thanks to intrinsic compliance or direct torque control [20], [21]. They excel at compliance but embed it implicitly with task gain tuning when using torque control, resulting in tracking errors once external forces intervene, or use high-gains joint position/velocity control to artificially deviate the robot's reference state from the reference trajectory, loosing intrinsic compliance of torque control.

Techniques of active compliance such as impedance control were extended to robots with harware elements of compliance such as the *Baxter* or *Sawyer* robots, shaping their interactive behaviour while exploiting passive compliance property. Although, these methods improve energy absorption while allowing controlability of the interaction behavior these methods inherits limitations from both passvie compliance and impedance control, unable to render high-stiffness tracking, or virtually high inertia [22].

For both optimization based or null-space projection based control, compliance, eapecially active compliance is present a particular application for physical interaction which is compliance in the null-space of a "main" task. Such application is shown in [6], which presents a strict hierarchy scheme for task-space control with compliance restricted to the null space. Their controller enforces the main task with high priority and projects an impedance-like behavior onto the redundant degrees of freedom, relying on disturbance observers such as residual estimation to estimate external forces. In contrast, our approach uses a unified quadratic programming formulation that does not rely on a strict hierarchy and can smoothly weight multiple tasks—both in the operational space and joint space—according to explicit compliance and performance demands. As a result, our framework not only retains stability and safety guarantees similar to [6], but also provides a more flexible and mathematically simpler mechanism for selectively tuning compliance or stiffness in different parts of the robot's motion, without reconfiguring the entire control structure. In addition, it remains robust to singularities through the use of low-priority regulation tasks.

Another recent approach [23], similar to ours, employs dynamical systems and impedance control for null-space compliance. While it benefits from a passivity proof, it does not account for safety or feasibility constraints. Consequently, their method could be incorporated as a task within our framework, thereby combining their passivity proof with our safety guarantees.

Impact-mitigation frameworks complement the above techniques by monitoring joint-torque residuals or distributed tactile skins and triggering evasive reactions upon collision [24],

[25], [26]. These methods can implement various reactive strategies to reduce peak forces after contact, but do not influence the nominal controller before the constraint violation occurs. Although essential for reacting to collisions, such schemes are inherently *reactive*: they activate only after an event has occurred.

Inverse-dynamics or whole-body controllers were formulated as Quadratic Programs to accommodate tasks with strict or weighted priority under hard joint, torque, and collision constraints [27] embedded directly into the torque reference [28] transmitted to the low-level joint torque control loop. Such method has also been extended to accommodate multiple actuated/non actuated, fixed/floating robots and their contact forces [29]. Nonetheless, apart from known modeled contacts, QP stacks mostly treat external forces as unknown disturbances, neglecting them or accommodating only via slack variables. Moreover, compliance is rarely an explicit objective; instead it appears as trajectory deviation, inherited from torque-control's non-modelled acceleration errors, which jeopardises constraint satisfaction under large disturbances.

Safety objectives are typically expressed either as constraints in optimization schemes [27] or as higher-priority objectives in null-space projection control schemes [30]. These objectives are often formulated at the velocity level, but are then adapted to the acceleration level to allow their use in inverse-dynamics control or second-order quadratic programs.

However, this conversion from velocity to acceleration is often handled inadequately, particularly in the choice of the gain. In many cases, the acceleration-level dynamics are defined so that the velocity-level target is reached in a single control iteration [27], [29], or, in more general forms, by simply setting the gain equal to the control frequency [30].

While high control frequencies are desirable to ensure safety limits are respected, such high gains in the safety-limit dynamics can also amplify sensor noise—especially in closed-loop torque control. This issue has been largely overlooked in previous work.

Finally, all above methods benefit fom timely and accurate external-force information, for which a variety of estimation methods and sensor-fusion schemes are now available [24], [31], [32], [33], [34]. F/T sensors, broad-area tactile skins offer direct but localised measurements [33], [34], are a heavy burden and costly, observers recover external wrenches from generalized-momentum/energy based residual [24], [31], [32] but suffer from delays and model uncertainties. No single modality yet provides timely, accurate force data over the whole structure and spectrum, which is crucial for compensating disturbances in the control loop itself, yet can help significantly reducing deviation and acceleration tracking problems of torque control when properly accounted for.

To summarize, despite their performance, current methods leave the following key gaps:

- **G1**: Guaranteeing safety constraints under external forces while remaining compliant when possible.
- **G2**: Smoothly transitioning between low-stiffness compliance and high-stiffness modes without causing chattering or abrupt movements when constraints (like joint or velocity limits) become active.

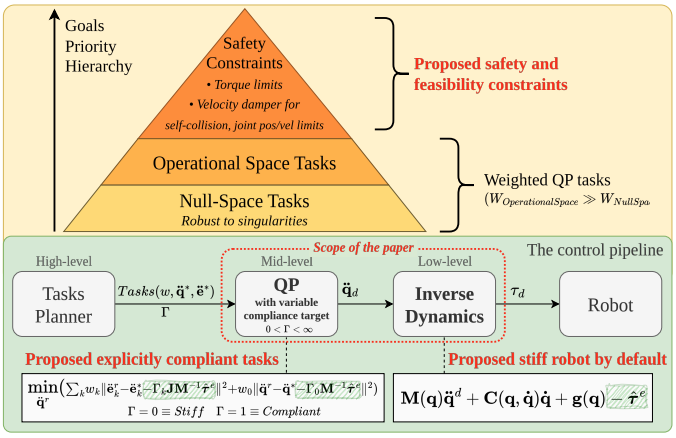


Fig. 1. Overview of the proposed control framework. A priority hierarchy (top) embeds safety constraints—joint/velocity limits, self-collision, and torque limits. The mid-level QP (green) fuses these constraints with explicit task compliance, while inverse dynamics compensates for the estimated external forces to execute the resulting commands.

- **G3**: Definition of the stiffness / compliance of any task in a flexible manner, regardless of its formulation or gains. Stiffer tasks showing higher accuracy and compliant tasks resulting in safer interaction forces.

B. Paper contributions and organization

This article expands on our RSS 2025 demonstration [35] by presenting the underlying theoretical foundations, detailing the methodology, and providing additional experimental validation to explain and justify the performance observed in that demonstration. In light of the gaps (**G1**, **G2** and **G3**) identified in the literature, we propose a torque-control framework summarized in Fig. 1 that offers the following contributions:

- **Robot stiffness definition based on safety constraints (G1)**: We design the framework such that all safety constraints (e.g. joint limits and collision avoidance) are "stiff" limits when they are in imminent danger of being violated. In doing so, the controller becomes stiff only when necessary, ensuring proper acceleration tracking therefore preventing safety breaches, while remaining potentially compliant otherwise.
- **Refined acceleration-level velocity damper (G2)**: We introduce an improved second-order velocity damper initially proposed in [36], fixing the choppiness that occurred in the case of the previous non-smooth formulation of [27]. We performed an experimental analysis of the effect of the proposed dynamics' gains, showing why the gains should not depend on the control frequency contrary to usual practices [30].
- **Flexible assignment of stiffness or compliance to different tasks (G3)**: We make it simple to define how stiff any task is, allowing also any combination of stiffness and compliance. For instance, we could leverage redundancy with *null-space compliance*—the robot remains stiff and accurate in a specified "main" task while remaining compliant in the dimensions that don't interfere with it. Rather than letting compliance appear as mere deviation from the task reference, our framework represents compliance as an *explicit* task objective that guides the robot to move along external forces.

• **Thorough experimental evaluation**: We use the Kinova Gen3 robot equipped with a force sensor at the wrist and perform several demonstrations of the guarantees, the performance and the versatility of our framework.

The remainder of this paper is organized as follows: Section II introduce background concepts on the targeted robot dynamics and tasks formulation, Section III describes our control framework, including details on the integration of safety constraints, Section IV presents the experimental setup and the comprehensive results, Section V discuss some aspects of the proposed method, and Section VI summarizes findings and suggests axes for future work.

II. Preliminary concepts

In this section, we describe the classic framework for torque control based on a quadratic program with the joint accelerations as decision variables, and in subsequent sections, we will present our proposed contributions. Our goal is to generate safe, feasible motions that accommodate the tasks planned by the high-level with the mid-level's safety requirements. To this end, we employ an *acceleration-based* quadratic program (QP), which enforces a good approximation of a structured hierarchy of priorities that govern how multiple objectives are handled while ensuring the feasibility and safety constraints (Section III-D). Within this framework, we use operational space tasks for the primary objectives (e.g., end-effector positioning) and exploit kinematic redundancy with null-space tasks to handle secondary objectives such as joint configuration regulation.

Robot dynamics: The scope of this paper targets serial manipulator robots; extensions can be made to other kinds of robots.

Let us define an n degrees of freedom (DoF) robot with joint configuration space $\mathcal{Q}$, represented by elements of $\mathbb{R}^n$. We call $\mathbf{q} \in \mathcal{Q}$ the joint configuration of the robot, $\dot{\mathbf{q}} \in \mathbb{R}^n$ the joint velocity, $\ddot{\mathbf{q}} \in \mathbb{R}^n$ the joint acceleration, and $\boldsymbol{\tau}_d \in \mathbb{R}^n$ the reference joint torques that is the input of the low-level torque regulation control loop.

The dynamic model of such a manipulator robot under external forces can be written as follows [37]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_d + \boldsymbol{\tau}_e \tag{1}$$

where $\mathbf{M}(\cdot) \in \mathbb{R}^{n\times n}$ and $\mathbf{C}(\cdot) \in \mathbb{R}^{n\times n}$ are the inertia and Coriolis matrices respectively such that $\dot{\mathbf{M}}(\cdot) = \mathbf{C}(\cdot) + \mathbf{C}(\cdot)^T$ and $\mathbf{g}(\cdot) \in \mathbb{R}^n$ is the gravity vector. Other elements such as friction torques can be added to this model, however, it is out of the scope of this paper, and we assume the low-level torque controller properly accounts for them, as guaranteed by the demonstrated low-level control in [35]. The external torques[1] $\boldsymbol{\tau}_e \in \mathbb{R}^n$ represents the external joint torques and defined as:

$$\boldsymbol{\tau}_e = \sum_{i=1}^{n_c} \mathbf{J}_i(\mathbf{q})^T \mathbf{F}_{e.i}, \tag{2}$$

where $\mathbf{F}_{e,i} \in \mathbb{R}^6$ are the i-th external force and moment, n_c is the number of external forces, $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times n}$ is the Jacobian of the contact point [6]. We define the joint acceleration deviation due to external forces as $\ddot{\mathbf{q}}_e = \mathbf{M}^{-1}\boldsymbol{\tau}_e$.

Operational Space Task: A Euclidean[2] task is commonly defined by an error function $\mathbf{e}_k(\mathbf{q}, t) = \mathbf{x}(\mathbf{q}) - \mathbf{x}^o(t)$ that we want to minimize, where $\mathbf{e}_k \in \mathbb{R}^m$, the subscript k denotes the task index and which doesn't necessarily reflect its priority, finally $\mathbf{x}^o(t)$ is the feedforward objective task position. Tracking this task is classically done by targeting converging second-order dynamics:

$$\ddot{\mathbf{e}}_k^* = -\lambda_{k,1}\dot{\mathbf{e}}_k(\mathbf{q}, \dot{\mathbf{q}}, t) - \lambda_{k,2}\mathbf{e}_k(\mathbf{q}, t) \tag{3}$$

where $\lambda_{k,1}$ and $\lambda_{k,2}$ are positive gains and $\ddot{\mathbf{e}}_k^*$ is the ideal error acceleration. We then must find feasible values of the task acceleration $\ddot{\mathbf{e}}_{d,k}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_d, t)$ that track those dynamics.

The following expression shows the affine mapping between $\ddot{\mathbf{e}}_{d,k}$ and $\ddot{\mathbf{q}}_d$:

$$\ddot{\mathbf{e}}_{d,k}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_d, t) = \mathbf{J}_k\ddot{\mathbf{q}}_d + \dot{\mathbf{J}}_k\dot{\mathbf{q}} - \ddot{\mathbf{x}}^o, \tag{4}$$

where $\ddot{\mathbf{x}}^o$ is the feed-forward objective acceleration of the task, and $\mathbf{J}_k$ is the Jacobian matrix of the k-th task. Note that this expression is linear in the decision variable $\ddot{\mathbf{q}}^r$. However, the second-order decay of (3) is not always feasible because of the higher-priority constraints or just because the task is out of the reachable space of the robot. A classic solution is then to minimize the second-order tracking error. In other words, we want the acceleration reference to be the solution to this optimization

$$\ddot{\mathbf{q}}_d = \arg\min_{\ddot{\mathbf{q}}^r}(\|\mathbf{J}_k\ddot{\mathbf{q}}^r + \dot{\mathbf{J}}_k\dot{\mathbf{q}} - \ddot{\mathbf{x}}^o - \ddot{\mathbf{e}}_k^*\|^2), \tag{5}$$

subject to feasibility and safety constraints. We track multiple tasks using weighted optimization:

$$\ddot{\mathbf{q}}_d = \arg\min_{\ddot{\mathbf{q}}^r}(\sum_k w_k\|\ddot{\mathbf{e}}_k^r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}^r, t) - \ddot{\mathbf{e}}_k^*\|^2), \tag{6}$$

where $\ddot{\mathbf{e}}_k^r = \mathbf{J}_k\ddot{\mathbf{q}}^r + \dot{\mathbf{J}}_k\dot{\mathbf{q}} - \ddot{\mathbf{x}}^o$ and w_k is the weight of the k-th task.

Null-Space Task: In the case of redundant robots, there is variability in the way a task can be achieved. This is an advantage because it provides better performance despite the activation of constraints. However, this variability may create joint position drift and discontinuities in the decision variable over time. Therefore, it is recommended to give as a final goal a reference to the acceleration in null space [38]. The simplest way to do it is by providing a reference value $\ddot{\mathbf{q}}^*$ to the joint acceleration itself with the lowest priority.

This reference value can be set to give a reference posture (posture task) or to act just like a damping. For convenience, we number this task 0, with the Jacobian being the identity matrix and w_0 being the weight.

Note that the hierarchy between the operational space tasks and this null-space task is set through weighting rather than strict constraints, which may slightly decrease the quality of the operational task tracking. Nevertheless, it allows the prevention of problems related to the divergence of $\ddot{\mathbf{q}}^*$ in case

[1] In this paper we assume external forces and moments to be applied by the environment on the robot, and not by the robot to the environment.

[2] The same principles apply in the case of orientations but expressed in $SO(3)$ and its Lie algebra

of task singularity while being generally a good approximation of hierarchy [39].

Therefore, the overall quadratic program has the following shape

$$\ddot{\mathbf{q}}_d = \arg\min_{\ddot{\mathbf{q}}^r} \quad (\sum_k w_k \|\ddot{\mathbf{e}}_k^r - \ddot{\mathbf{e}}_k^*\|^2 + w_0 \|\ddot{\mathbf{q}}^r - \ddot{\mathbf{q}}^*\|^2); \quad (7)$$
$$\text{such that} \quad \mathbf{A}_c \ddot{\mathbf{q}}^r \leq \mathbf{b}_c,$$

where $\mathbf{A}_c$ and $\mathbf{b}_c$ gather all the feasibility and safety constraints and $\ddot{\mathbf{q}}^*$ is the desired joint acceleration. Additional details on the classic implementation of this QP motion solver with torque-control are available in [40].

III. Proposed framework

A. Objective of the framework

In this framework, we want to compute the robot's reference joint torques $\boldsymbol{\tau_d}$ from a reference joint acceleration $\ddot{\mathbf{q}}_\mathbf{d}$. We use a QP to compute the reference joint acceleration so that it fulfills the hierarchically sorted objectives illustrated in the top section of Fig. 1, detailed below in descending order of priority.

- Safety and feasibility, which ensures that the generated motion adheres to a set of constraints that aims to keep the reference motion within the boundaries or the robot's capabilities and mitigate risks by limiting those capabilities if necessary.
- Operational space task(s), these are the main tasks to be achieved by the control, commonly the 3D Cartesian pose of an end-effector. This could also be a set of weighted tasks or multiple levels of hierarchically ordered tasks.
- Null-space task, it is a task responsible for handling the possible infinity of ways the operational space task could be achieved. This task is necessary for redundant robots, when the dimension of the space covered by operational space tasks is less than the robot's number of DoF. This happens for example when a 6 DoF robot needs to track only the 3D Cartesian end-effector position, leaving 3 DoF of redundancy.

B. Inverse Dynamics with External Force Compensation

One key idea of the proposed framework is to enforce a proper tracking of the reference accelerations on the real robot under external forces, such that the reference acceleration ensuring safety constraints or precise tasks is ensured, otherwise made compliant explicitly. This means on the one hand that the robot needs to be stiff by default, negating the effects of external forces and ensuring tracking of the acceleration, and on the other hand that the tasks must include their compliance effects explicitly in the desired acceleration formulation, as explained in Section III-C. To do so, we calculate an estimate of the external torques $\hat{\boldsymbol{\tau}}_e$, explained in Section **??**, and compute the commanded torque, sent to the robot's low-level torque controller, as follows

$$\boldsymbol{\tau}_d = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) - \hat{\boldsymbol{\tau}}_e, \quad (8)$$

Unlike the baseline QP torque controller that doesn't account for external-forces, with a good estimation of the external forces, the redefinition of (8) prevents the robot from moving due to external forces, no matter how small the task gains is. Indeed, when combined with the dynamic equation (1), under the assumption that $\hat{\boldsymbol{\tau}}_e = \boldsymbol{\tau}_e$, we achieve the ideal joint acceleration tracking $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d$ independently of external forces.

Since through (8) accurate acceleration tracking is enforced, it's necessary to express compliance in the desired acceleration $\ddot{\mathbf{q}}_d$. The next section explains how we define compliance explicitly as an objective in the QP.

C. Explicit Formulation of Compliance in Task Objectives

To embed compliance in the desired acceleration, it needs to be part of the QP's objective, instead of defining a compliance objective that aims at minimizing forces applied onto the robot independently of the tasks, we formulate compliance as part of the task, by introducing an artificial deviation of the tasks reference acceleration equal to $\ddot{\mathbf{q}}_\mathbf{e}$, the acceleration deviation equivalent to external-forces. Thanks to this formulation, we can select independently the compliant behavior for each task, which is also accounted in tasks hierarchy. This is done by replacing the QP in (7) with the following new formulation of the optimization problem

$$\min_{\ddot{\mathbf{q}}^r} (\textstyle\sum_k w_k \|\ddot{\mathbf{e}}_k^r - \ddot{\mathbf{e}}_k^* - \Gamma_k \mathbf{J}\hat{\dot{\mathbf{q}}}_e\|^2 + w_0 \|\ddot{\mathbf{q}}^r - \ddot{\mathbf{q}}^* - \Gamma_0 \hat{\dot{\mathbf{q}}}_e\|^2);$$
$$\text{such that} \quad (9)$$
$$\mathbf{A}_c \ddot{\mathbf{q}}^r \leq \mathbf{b}_c,$$

where $\Gamma_k \in \mathbb{R}_+^{m \times m}$ is a semi-positive definite matrix defining the desired compliance of the k-th task with m the task's dimensions and $\ddot{\mathbf{e}}_k^f = \mathbf{J}\hat{\dot{\mathbf{q}}}$ is the acceleration deviation in the task space due to the effect of the external force.

By setting $\Gamma_k = \mathbb{I}_m$, the desired task acceleration becomes $\ddot{\mathbf{e}}_k^* - \ddot{\mathbf{e}}_k^f$, which means that we explicitly require the task to deviate according to the influence of the external forces, and thus to be compliant. Similarly, Γ_0 defines the compliance of the posture acceleration task, which we use to regulate the compliance in the null-space of the task.

Section IV-E experimentally explores the effect of the compliance parameter Γ on different compliance setups.

D. Proposed Safety and Feasibility Constraints

1) Torque Limits: We implement the following torque limits on the robot's joints:

$$\boldsymbol{\tau}^- \leq \boldsymbol{\tau}_d \leq \boldsymbol{\tau}^+, \quad (10)$$

where $\boldsymbol{\tau}^-$ and $\boldsymbol{\tau}^+$ represent the lower and upper bounds for the joint torques, and $\boldsymbol{\tau}_d$ is the reference joint torque defined in (8) which linearly depends on the decision variable $\ddot{\mathbf{q}}_d$[3].

This prevents the QP from generating unfeasible accelerations when the robot is subjected to strong forces, maintaining

[3]Note that with this constraint formulation and (8), the torque limit constraint accounts for the external torque, which is not the case in classic formulations.

optimal accelerations with respect to tasks despite external forces and torque limits, which is also valid with all the other constraints.

2) Velocity Damper Formulation for Closed-Loop Inverse Dynamics Controllers: In addition to the torque limits, we enforce kinematic constraints such as joint position/velocity limits and (self-) collisions. In this section, we introduce a close-loop reformulation of the velocity damper presented in [36] for open-loop controllers.

Regarding the joint position/velocity constraints, we present hereinafter the method to ensure the upper limits, which can be easily extended to the lower ones.

As the constraint is expressed similarly for all the joints, we will omit the joint index and refer to the joint position as non-bold q. We define the limits as:

$$q \leq q^+ \tag{11}$$
$$\dot{q} \leq \dot{q}^+ \tag{12}$$

where the superscript $^+$ represents the upper limit of both the joint position q and velocity $\dot{q}$, respectively.

To present our approach, we first remind the method introduced in [36] and its adaptation in [27], and its limitations when applied on closed-loop torque control, and we present the modifications allowing to overcome them.

Matching the notations of these previous works, we define the joint position distance to the upper limit $e = q^+ - q$, a safety distance d_s to that limit, and the upper velocity limit $\dot{e}_{\min} = -\dot{q}^+$.

The resulting constraints are the following:

$$e \geq d_s \tag{13}$$
$$\dot{e} \geq \dot{e}_{\min} \tag{14}$$

However, since these constraints are not expressed in terms of the decision variable $\ddot{\mathbf{q}}^r$ they are adapted to be inserted in the QP, by defining a limit in acceleration that guarantee that the constraint is respected.

The first step is to create an intermediary velocity limit $\dot{e}_d$

$$\dot{e}_d = -\xi \frac{e - d_s}{d_i - d_s} \tag{15}$$

where ξ is a positive constant gain that controls the maximal allowed convergence speed and d_i is a constant influence distance (threshold where this constraint is inserted in the QP). Therefore, respecting the constraint

$$\dot{e} \geq \dot{e}_d \tag{16}$$

guarantees that the position limit will be respected with the defined safety margin. With equations (14) and (16), both joint position and velocity constraints become velocity limits.

The literature provides the following adaptation for joint acceleration control [27] by using finite differences derivation approximation:

$$\ddot{e}_p = -\frac{1}{\delta t} \cdot (\dot{e} - \dot{e}_d) \tag{17}$$
$$\ddot{e}_v = -\frac{1}{\delta t} \cdot (\dot{e} - \dot{e}_{\min}) \tag{18}$$

where δt is the control sampling time; $\ddot{e}_p$ and $\ddot{e}_v$ are defined to respectively denote the joint position and the velocity limits in acceleration. These constraints can be merged into one reference acceleration using $\max$ function:

$$\ddot{q}_d < -\max(\ddot{e}_p, \ddot{e}_v) \tag{19}$$

Since the constraint depends linearly on the decision variables and can be added as a constraint in the QP.

To address issues w.r.t. to the formulation of constraint dynamic we propose a new formulation of the velocity damper that eliminates the δt dependency. First, we redefine the limit acceleration variable of (17) with a smaller positive damping gain λ, as follows:

$$\ddot{e}_p = -\lambda(\dot{e} - \dot{e}_d). \tag{20}$$

To properly determine λ, we rearrange (20) with (15) to identify a second-order system, where $\ddot{e}_p := \ddot{e}$. This gives:

$$\frac{K}{\lambda}\ddot{e} + K\dot{e} + (e - d_s) = 0 \tag{21}$$

where $K = \frac{d_i - d_s}{\xi}$.

This allows to identify the equivalent second-order dynamics of the Laplace transform of the variable $(e - d_s)$:

$$\tau_s^2 s^2 + 2\zeta\tau_s s + 1 = 0 \tag{22}$$

where s is the Laplace transform variable, $\tau_s = \sqrt{\frac{K}{\lambda}}$ is the time constant, and $M = \zeta = \frac{\sqrt{\lambda K}}{2}$ is the damping factor of the second order system.

Because this system is a second-order system, dynamic of the constraint may overshoot the limit, to ensure the constraint it is imperative to ensure at least a critically damped response ($M \geq 1$) theoretically providing the fastest convergence. However, in practice, measurement noise, numerical precision limitations, and unmodeled dynamics can push the system away from its reference. Therefore, to ensure robustness, we enforce an overdamped dynamics with $M > 1$, preventing oscillations and better ensuring constraint enforcement. From this system definition, we obtain:

$$\lambda = \frac{4M^2\xi}{d_i - d_s} \tag{23}$$

Then, we can write $\ddot{e}_p$ in function of λ and M:

$$\ddot{e}_p = -\lambda\dot{e} - \frac{\lambda^2}{4M^2}(e - d_s) \tag{24}$$

For consistency, we use the same gain for the velocity limit for which the constraints can be merged and added to the QP with (19).

Lower constraints can be set similarly by setting $e = q - q^-$ and $\dot{e}_{\min} = \dot{q}^-$.

This new formulation of the acceleration-based velocity damper eliminates the δt dependency, allowing the damping gain λ to be freely selected. Additionally, the introduced M parameter enables explicit control over the damping level, ensuring that constraint enforcement remains robust even in high-frequency closed-loop systems.

The (self-) collision constraint is implemented similarly to the approaches described in [27] and [41], where they

leverage similar formulation as the velocity damper but with the e variable describing the distance between the closest points on the body pair. In this case also, we apply the new formulation of the velocity damper instead of the δt-dependent formulation.

A comparative analysis between the δt-dependent formulation and our proposed approach is conducted in Section IV-B through experimental validation. Furthermore, the impact of varying the damping parameter λ and the overdamping factor M is also evaluated.

IV. Experimental validation

A. General experimental setup

1) Hardware: Experiments were conducted on a Kinova Gen 3, a 7-DoF manipulator with four limitless joints, interfaced through our open-source framework[4] based on *mc_rtc*[5] and Kinova's *Kortex* API at 1 kHz. A Bota SensONE 6-axis F/T sensor was mounted at the end-effector and connected via Ethernet to a NUC through *EtherCAT*. Measurements were published through ROS and retrieved within 1 ms by the control PC.

2) QP control: The controller is implemented in *mc_rtc*, which formulates tasks and safety constraints as a QP that solves for $\ddot{\mathbf{q}}^r$. The desired torque $\boldsymbol{\tau}_d$ is then computed using (8) and sent to the robot via our *mc_rtc*–Kortex interface.

3) External force estimation: External joint torques were obtained by combining F/T sensor measurements, $\boldsymbol{\tau}_s = \mathbf{J}_s^T \mathbf{F}_s$, with estimates from a momentum observer [24]:

$$\mathbf{r} = \mathbf{K}_I \left[\mathbf{p}(t) - \int_0^t (\boldsymbol{\tau} + \mathbf{C}^T \dot{\mathbf{q}} - \mathbf{g} + \mathbf{r}) ds - \mathbf{p}(0) \right]. \quad (25)$$

Because the residual behaves as a low-pass filter, the two sources are fused through a complementary filter:

$$\boldsymbol{\tau}_e = \mathbf{r} + \frac{s}{K_I + s} \boldsymbol{\tau}_s, \quad (26)$$

where s is the Laplace variable. This ensures accurate low-frequency estimation from the observer and high-frequency contribution from the sensor.

B. Velocity-Damper Comparison Results

We show experimental results of the novel velocity-damper formulation for multiple combinations of the parameters M and λ, in a scenario where both velocity and joint-position constraints are triggered.

To evaluate the novel formulation of the velocity-damper (24) described in Section III-D2, we tested the joint velocity and the joint limit constraints. While both joint position limit and collision constraints rely on the 2nd order velocity damper, we chose to focus on the joint limit constraint for its greater experimental repeatability. The safety distance for each joint is set to $d_s = 0.01(q^+ - q^-)$, corresponding to 1% of the joint's total range. In the case of infinite joints the position constraint doesn't apply. Since we rely on (24) for the constraint's dynamics, it is not required to define d_i, and therefore, it is not used as a constraint activation threshold. We design a repeatable experimental setting where we use a 1.25 kg weight fixed to the end-effector. We position the robot using high gain position control in an L-shape such that only the 4th joint (elbow) is influenced by gravity. Once the robot is positioned, we switch the robot to torque control and make the 4th joint compliant to external forces, by setting only the 4th element of the diagonal matrix Γ_0 to 1 while keeping the rest to 0. It results in the 4th joint rotating downward in the direction of the gravity towards the joint's limit. We used external force feedback and kept the other joints non-compliant to maintain the robot's posture consistently across tests even when reaching constraints.

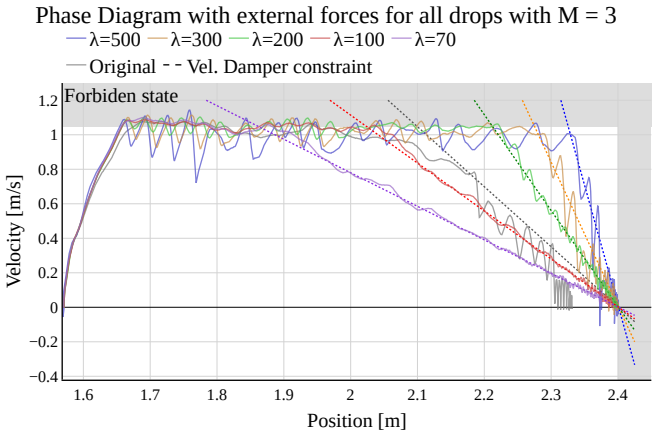


Fig. 2. Phase diagram for all drop sequences where $M = 3$ including feedback on external forces. The *x-axis* represents the position in meters, and the *y-axis* represents the velocity in meters per second. Each line corresponds to a different value of λ, with colors indicating different λ values as specified in the legend. The *grey* line represents the case where we used the velocity damper as formulated in [27] for comparison. The shaded regions indicate forbidden states from software limits where the system aims to ensure operation in those limits. The dashed lines represent the velocity limit and velocity damper constraint.

We demonstrate the effect of M and λ values on the behavior of both joint velocity limit and joint position limit. For all results, we also included a case that used the original formulation of the velocity-damper in *grey*.

Fig. 2 shows the phase diagram of multiple drops, which use the same value of $M = 3$ to show the effect of λ. We chose the value of $M = 3$ only for the sake of the diagram's readability. This plot shows how for high values of λ (e.g. $\lambda = 500$ in *blue* or $\lambda = 300$ in *yellow*) the system produces large oscillations compared to lower yet reasonable values (e.g. $\lambda = 100$ in *red*, $\lambda = 70$ in *purple*).

Fig. 3 shows this time the phase diagram of multiple drops, which use the same value of $\lambda = 100$ to show the effect of M. Again, we chose the value of $\lambda = 100$ only for the sake of the diagram's readability as it has been shown to produce much lower oscillations. This plot shows how for values of M close to 1 ($M = 1.5$ in *purple*), the velocity damper slows the system closer to the limit but with higher deceleration, which requires higher torques.

[4]https://github.com/mathieu-celerier/mc-kinova-sim-superbuild
[5]https://jrl-umi3218.github.io/mc_rtc/

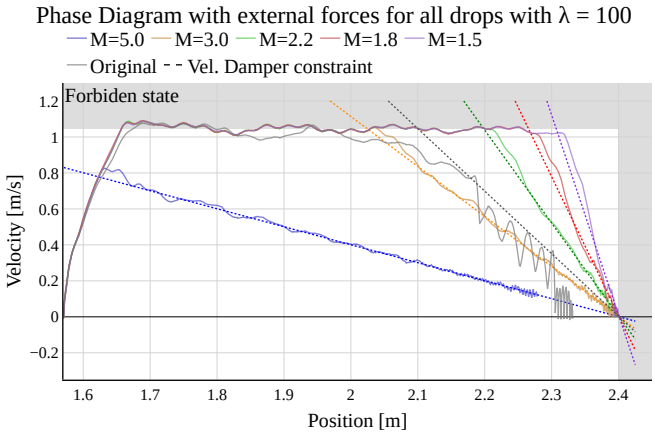


Fig. 3. Phase diagram for all drop sequences where $\lambda = 100$ including feedback on external forces. The *x-axis* represents the position in meters, and the *y-axis* represents the velocity in meters per second. Each line corresponds to a different value of M, with colors indicating different M values as specified in the legend.

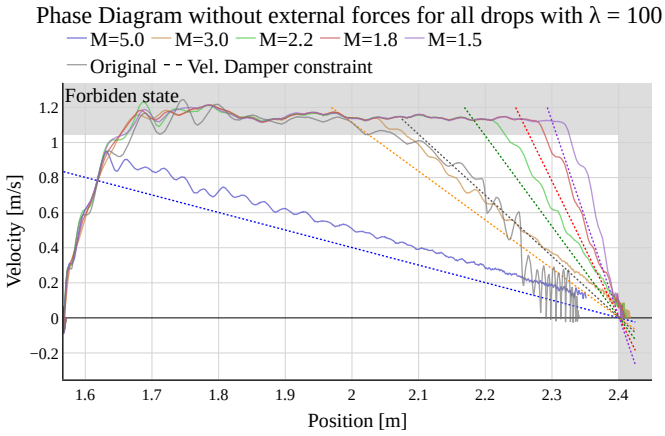


Fig. 4. Phase diagram for all drop sequences where $\lambda = 100$ and no feedback on external forces. The *x-axis* represents the position in meters, and the *y-axis* represents the velocity in meters per second. Each line corresponds to a different value of M, with colors indicating different M values as specified in the legend.

From those results, we recommend users of the novel formulation of the velocity damper first set a value of λ such that no oscillations appear, then choose the appropriate M that starts slowing the system not too far from the limit to prevent constantly damped system, but not too strongly as well. In our setup, we use $\lambda = 100$ and $M = 2$

C. Safety constraint under external forces

With the same setup, we studied the effect of explicitly considering the external forces on safety constraints.

We tested both the joint velocity limit and joint position limit with a similar experimental protocol in IV-B, the only difference being that we deactivated the feedback on the external forces. Again, we included in *grey* a comparison with the original formulation of the velocity damper.

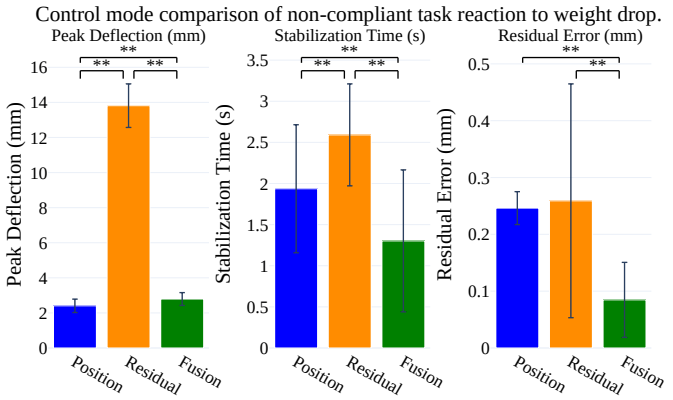


Fig. 5. Replication of the "mass-drop" experiment introduced in our RSS 2025 demonstration [35]. For the three performance metrics—**peak deflection**, **stabilisation time**, and **residual error**-we compare classical *Position* control (blue), *Torque + Residual* (orange), and the proposed *Torque + Fusion* (green). Bars show mean ± SD over 30 repetitions; brackets report pairwise significance from a Mann-Whitney U test (* $p < 0.05$, ** $p < 0.001$).

Fig. 4 shows the phase diagram of multiple drops without external forces feedback, which uses the same value of $\lambda = 100$ as in Fig. 3. We can see that regardless of the combination of (M, λ) when there is no feedback on external forces, the robot's state overshoots all the limits/constraints—entering potentially dangerous states with too high velocity or exceeding position/collision limits, which could result in pinching or crushing of body parts. Therefore, by comparing Fig. 3, where external force estimation is incorporated to account for the constraints, with Fig. 4, where it is not, it is evident that incorporating feedback from external force estimation helps maintaining the system within the constraints. In contrast, when this feedback is removed, the system clearly exceeds the constraint limits.

D. Explicit non-compliance performance

We reproduced the "mass-drop" experiment presented in [35]. Starting from the nominal posture, the end-effector is extended 5 cm forward to increase the moment arm and reduce joint-friction influence. A 1.25 kg mass is attached at the wrist, lifted to the attachment point, and released. Each control mode is tested 30 times, with 4 s of data analysed from the instant of release.

During torque control the task space (pose) is defined as non-compliant while the null-space remains compliant. For position control we implemented an acceleration-based PID to mimic the (assumed) internal loop of the Kinova Gen3, ensuring a fair baseline.

Figure 5 shows that, by including feedback on the external forces with precise measurements in *green*, peak deflection is similar to position control while both stabilization time and residual are smaller, therefore we achieve better performance than position control. By combining residual force estimation with direct external-force measurements, we achieved instantaneous compensation, eliminating the need for deviation of joint positions to generate reaction forces as in position control. The main limitations of these results lie in the robot flexibility

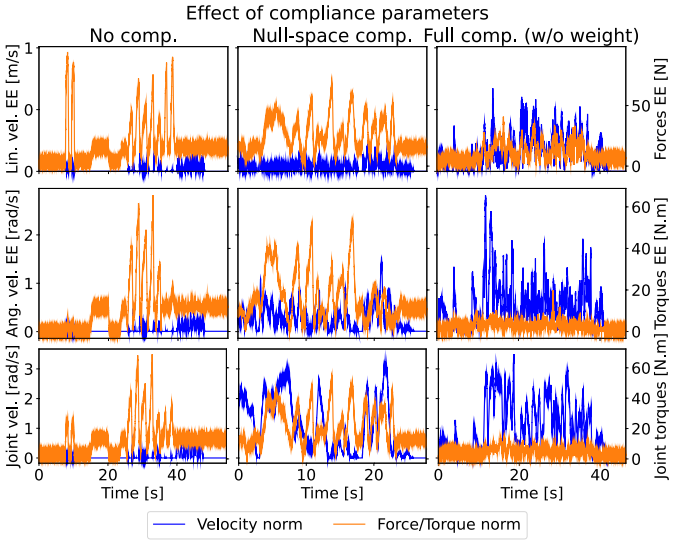


Fig. 6. Measurement of 6D Cartesian and posture task velocities (*blue*) and the external forces/torques (*orange*) applied to them under three compliance setups. In the non-compliance setup, both tasks are stiff with $\Gamma = \mathbf{0}$. In the null-space compliance setup, only the posture task is compliant with $\Gamma = \mathbb{I}_7$. In the full compliance setup, both tasks are compliant with $\Gamma = \mathbb{I}$, allowing for total robot manipulation.

and a better tracking of the reference by the robot's low-level control such that the exact required reaction force can be properly applied to the environment.

Because the experimental conditions, apparatus, and statistical treatment are identical to our paper we expands on [35], readers can view Fig. 5 as a direct, one-for-one extension of Fig. 3 in [35]. The improvements here therefore quantify the benefit brought *solely* by the new compliance formulation and the fusion estimator, validating the framework's capacity to combine precision with safety-critical compliance.

E. Compliance

Because our method explicitly estimates and compensates for external forces, we must specify which tasks should remain compliant. Without such specification, all external forces would be fully compensated, suppressing compliant behavior at the task level.

To illustrate the effect of compliance selection, we use the proposed approach with the robot in its default configuration. We define:

- A **6D Cartesian task** to fix the end-effector's position and orientation. This task is non-compliant by default.
- A **posture task** with lower priority in the QP hierarchy (null-space task) and significantly lower weight.

By adjusting the compliance selection matrix Γ, we can produce different behaviors:

1) **Fully non-compliant control**: $\Gamma = \mathbf{0}$ for both tasks. External forces are entirely compensated, resulting in minimal robot motion.
2) **Null-space compliance**: $\Gamma = \mathbb{I}_7$ for the posture task only. External forces are still compensated at the end-effector but cause motion in joint space, enabling compliance in the null space.
3) **Full compliance**:$\Gamma = \mathbb{I}_6$ for the Cartesian task and $\Gamma = \mathbb{I}_7$ for the posture task. The robot responds freely to external forces in all degrees of freedom.

To test these different compliance modes, a 1.25 kg weight was attached to the F/T sensor at the end-effector using an 18 cm rope. Additional external forces were manually applied during the experiments. The weight was removed for the full compliance condition, as it caused the robot to collapse due to the absence of compensatory torques.

Fig. 6 illustrates the effect of the different compliance configurations by showing the norm of the linear and angular velocity of the 6D Cartesian task, along with the norm of the joint velocities. These metrics highlight how the robot reacts to external forces under each compliance condition. The results confirm that non-compliant control suppresses motion, null-space compliance enables joint-level adaptation, and full compliance allows free manipulation of the robot in all directions. These compliance scenarios are demonstrated in the *supplemental material 1* video and shows how solely through the compliance parameter with fixed task gains it's possible to achieve very different compliance behavior.

V. Discussions

While this approach introduces an additional parameter to configure, it grants the flexibility to determine whether a control task should be compliant, and to what extent, across different scenarios. When the compliance parameter is set to the identity matrix, the method reduces to classical inverse-dynamics control, if combined with standard PD dynamics, it reproduces nominal impedance control. More generally, any combination of compliance and task stiffness can be specified. Advanced tuning of Γ_k further increases flexibility: its eigenvalues define directional stiffness along their eigenvectors, enabling anisotropic compliance, as illustrated in *supplemental material 2*. These eigenvalues also determine the ratio between intrinsic and effective inertia during interaction: zero implies infinite effective inertia, one corresponds to matched (classical) compliance, and other positive values allow fine-grained shaping of the compliance profile—though values far from one increase sensitivity to errors in external force estimation.

Explicit compliance also enables simulating different inertia matrices by modifying the estimated acceleration response to external forces. By combining Γ with any virtual inertia matrix, the robot can replicate a wide range of interactive behaviors, as demonstrated in *supplemental material 3*, where it reproduces the dynamics of a door with gravity and friction effects.

When all tasks are fully compliant ($\Gamma_i = \mathbf{I}$), the compliance term cancels out the external torque compensation in (8), yielding behavior identical to a torque-controlled robot that ignores external forces. In this case, the method is no more sensitive to force-estimation errors than classical torque control. However, once a safety constraint becomes active, the proposed method ensures correct treatment of external forces, as shown in Fig.3, in contrast to the failures of classical approaches in Fig.4. This demonstrates the robustness and safety of the proposed control strategy.

VI. Conclusion

In this paper, we introduced a new paradigm of compliant control for torque-controlled robots by incorporating an explicit compliance definition directly into the QP-based optimization problem. This approach ensures stiff respect for safety constraints through the compensation of external torques in the inverse dynamics. In addition, we defined torque limits by including the external torques applied to the robot, thus accounting for interactions in the safety constraints.

Regarding velocity damping, we proposed a new formulation in terms of accelerations that is independent of control frequency, making the resulting position and velocity constraints less noisy and easier to tune through the introduction of the gains λ and M. We anticipate that our method can be extended to other controllers, such as hierarchical QP solvers and null-space projection-based approaches.

Our approach, which combines force/torque sensor readings with residual-based estimation, has demonstrated better performance than using residual estimation alone, especially when providing feedback for external torque compensation. It enhances the bandwidth of external force measurements and maintains force estimation at any contact point. This may prove beneficial in human-robot interaction settings, ergonomics, and user perception.

While our method secures a foundational set of constraints, task-level extensions—such as task scaling—could enhance the robot's behavior during deviations from target trajectories, mitigating unpredictable motion under external perturbations.

Furthermore, our method could be applied to robots with poor torque tracking by building a reference acceleration that directly complies with external forces, offering an explicitly compliant trajectory. This capability could improve kinematic tracking for stronger compliance perception, especially when forces are measured through a force/torque sensor. Future research may incorporate force limit constraints for safer stiff tasks, explore hierarchical control for improved management of feasibility constraints, and adapt the framework for floating-base robots (e.g., legged robots) to address contact force constraints and friction considerations.

Similarly, other broader research axes leveraging the proposed method are of interest:

- **Generalization of Explicit Compliance:** Our experiments demonstrated explicit compliance for tasks and joints under torque control (using Γ_0 and Γ_k). However, because explicit compliance is formulated within the QP optimization and is valid as long as the robot is rendered stiff by default, it could be used with other control modes such as open-loop position control were only the external-forces are feedback to the mid-level.
- **Improvement of External Torque Estimation:** Although momentum-based residuals are common for external torque estimation, recent advances in super-twisting momentum-based observers [42] show promise for improved performance. Integrating a super-twisting-based observer for force/torque sensor fusion may enhance external torque estimation by avoiding the low-pass filter limitations of classic momentum-based approaches.

References

[1] L. E. Kavraki and S. M. LaValle, *Motion Planning*. Cham: Springer International Publishing, 2016, pp. 139–162. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_7

[2] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," vol. 55, pp. 248–266, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415818300321

[3] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *Int. J.of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[4] N. Hogan, "Impedance control: An approach to manipulation: Parts i-iii," vol. 107, no. 1, pp. 1–24, 1985. [Online]. Available: https://doi.org/10.1115/1.3140713

[5] A. Calanca, R. Muradore, and P. Fiorini, "A review of algorithms for compliant control of stiff and fixed-compliance robots," *IEEE/ASME Trans.on Mechatronics*, vol. 21, no. 2, pp. 613–624, 2016.

[6] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Trans.on Robotics*, 2013.

[7] M. Benallegue, R. Cisneros, A. Benallegue, A. Tanguy, A. Escande, M. Morisawa, and F. Kanehiro, "On compliance and safety with torque-control for robots with high reduction gears and no joint-torque feedback," in *IEEE/RSJ IROS*. IEEE, 2021.

[8] D. Kortenkamp, R. Simmons, and D. Brugali, *Robotic Systems Architectures and Programming*. Cham: Springer International Publishing, 2016, pp. 283–306.

[9] I. O. for Standardization, "Iso standard no. 10218," Robots and robotic devices — Safety requirements for industrial robots, Part 1: Robots, 2011, retrieved from https://www.iso.org/standard/51330.html.

[10] ——, "Iso standard no. 10218," Robots and robotic devices — Safety requirements for industrial robots, Part 2: Robot systems and integration, 2011, retrieved from https://www.iso.org/standard/41571.html.

[11] ——, "Iso standard no. 15066," Robots and robotic devices — Collaborative robots, 2016, retrieved from https://www.iso.org/standard/62996.html.

[12] ——, "Iso standard no. 13849," Safety of machinery — Safety-related parts of control systems, Part 1: General principles for design, 2016, retrieved from https://www.iso.org/standard/69883.html.

[13] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, "Model predictive robot-environment interaction control for mobile manipulation tasks," in *IEEE ICRA*. IEEE, 2021.

[14] M. Schumacher, J. Wojtusch, P. Beckerle, and O. von Stryk, "An introductory review of active compliant control," vol. 119, pp. 185–200, Sept. 2019.

[15] R. V. Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber, "Compliant actuator designs," *IEEE Robotics & Automation Magazine*, 2009.

[16] M. Polic, M. Car, F. Petric, and M. Orsag, "Compliant plant exploration for agricultural procedures with a collaborative robot," *IEEE Robotics and Automation Letters*, 2021.

[17] R. Natarajan, G. L. Johnston, N. Simaan, M. Likhachev, and H. Choset, "Torque-limited manipulation planning through contact by interleaving graph search and trajectory optimization," pp. 8148–8154, 2023.

[18] Z.-J. Li, H.-B. Wu, J.-M. Yang, M.-H. Wang, and J.-H. Ye, "A position and torque switching control method for robot collision safety," *Intl. J.of Automation and Computing*, 2018.

[19] Z. He, Y. Guo, T. Huang, and W. Liu, "Dynamical systems based compliance control approach without direct force information," in *IEEE Int. Conf. on Industrial Technology*, March 2024, pp. 1–7.

[20] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem, "Active compliant motion: a survey," *Advanced Robotics*, 2005.

[21] E. Magrini, F. Flacco, and A. De Luca, "Control of generalized contact motion and force in physical human-robot interaction," in *IEEE ICRA*. IEEE, 2015.

[22] C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "On the Passivity-Based Impedance Control of Flexible Joint Robots," vol. 24, no. 2, pp. 416–429, Apr. 2008.

[23] Z.-Q. Yang, M. Wang, and M. R. Kermani, "A null space compliance approach for maintaining safety and tracking performance in human-robot interactions," 2025. [Online]. Available: https://arxiv.org/abs/2502.02443

[24] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *IEEE/RSJ IROS*. IEEE, 2006.

[25] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans.on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[26] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[27] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, *et al.*, "Multi-contact vertical ladder climbing with an hrp-2 humanoid," *Autonomous Robots*, 2016.

[28] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans.on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2004.

[29] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic Programming for Multirobot and Task-Space Force Control," vol. 35, no. 1, pp. 64–77, Feb. 2019.

[30] F. Flacco, A. De Luca, and O. Khatib, "Control of Redundant Robots Under Hard Joint Constraints: Saturation in the Null Space," vol. 31, no. 3, pp. 637–654, June 2015.

[31] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *IEEE-RAS Humanoids*. IEEE, 2016.

[32] G. Garofalo, N. Mansfeld, J. Jankowski, and C. Ott, "Sliding mode momentum observers for estimation of external torques and joint acceleration," in *Int. Conf. on Robotics and Automation*, 2019, pp. 6117–6123.

[33] T. Hoshi and H. Shinoda, "Robot skin based on touch-area-sensitive tactile element," in *IEEE ICRA*. IEEE, 2006.

[34] G. Cheng, E. Dean-Leon, F. Bergner, J. Rogelio Guadarrama Olvera, Q. Leboutet, and P. Mittendorfer, "A comprehensive realization of robot skin: Sensors, sensing, control, and applications," *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2034–2051, 2019.

[35] B. Muraccioli, M. Celerier, M. Benallegue, and G. Venture, "Demonstrating a Control Framework for Physical Human-Robot Interaction Toward Industrial Applications," in *Robotics: Science and Systems (RSS) 2025*. Los Angeles, United States: RSS Foundation, June 2025.

[36] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *IEEE ICRA*. IEEE, 1987.

[37] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *IEEE/RSJ IROS*. IEEE, 2016.

[38] S. Chiaverini, G. Oriolo, and A. A. Maciejewski, *Redundant Robots*. Cham: Springer International Publishing, 2016, pp. 221–242. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_10

[39] K. Bouyarmane and A. Kheddar, "On weight-prioritized multitask control of humanoid robots," *IEEE Trans.on Automatic Control*, 2017.

[40] R. Cisneros, M. Benallegue, A. Benallegue, M. Morisawa, H. Audren, P. Gergondet, A. Escande, A. Kheddar, and F. Kanehiro, "Robust humanoid control using a qp solver with integral gains," in *IEEE/RSJ IROS*. IEEE, 2018.

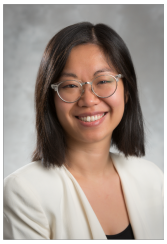
[41] F. Kanehiro, M. Morisawa, W. Suleiman, K. Kaneko, and E. Yoshida, "Integrating geometric constraints into reactive leg motion generation," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 4069–4076.

[42] J. Jiang, M. Lv, and J. Zhang, "An improved super-twisting sliding mode momentum observer for estimating the external torque of the collaborative robot," in *2024 43rd Chinese Control Conference (CCC)*, 2024, pp. 4615–4620.

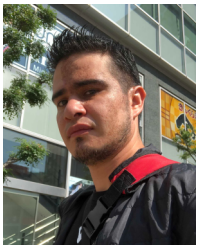
Mathieu Celerier Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.

Bastien Muraccioli Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.

Mehdi Benallegue Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.



Yue Hu (Member, IEEE) Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.



Rafael Cisneros-Limón Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.

Hiroshi Kaminaga Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.

Gentiane Venture (Senior Member, IEEE) Author biography placeholder. This text serves only as a placeholder for the author's real biography. The actual biography will be added in the final version.