On compliance and safety with torque-control for robots with high reduction gears and no joint-torque feedback

Mehdi Benallegue, Rafael Cisneros, Abdelaziz Benallegue, Arnaud Tanguy, Adrien Escande, Mitsuharu Morisawa, Fumio Kanehiro

On compliance and safety with torque-control for robots with high reduction gears and no joint-torque feedback

Mehdi Benallegue, Rafael Cisneros, Abdelaziz Benallegue, Arnaud Tanguy, Adrien Escande, Mitsuharu Morisawa, Fumio Kanehiro

***Abstract*— In this paper we report the safety-oriented framework for controlling the torque in the case of robots with high reduction gears and having no joint torque feedback. This kind of robots suffer from high joint friction and low backdrivability, requiring high gains and integral feedback, which can be dangerous. Our optimization-based framework includes feasibility and safety features borrowed from position control, and we introduce novel ones. We show how we limit the integral terms using a QP-based anti-windup which produces the optimal torque that maintains the best performances under safety limits. We show also a new controller for null-space compliance, providing strong guarantees of convergence in the task-space and ignoring the corresponding null-space where the robot can be moved freely. We validate these features with experiments on one 9 DoF arm of the robot HRP-5P performing a Cartesian task, and then a dual Cartesian / admittance task.**

I. Introduction

The interaction and proximity between humans and robots is often considered today as a last resort in industrial contexts which still prefer separated environments. Like this Japanese company which, as acknowledged by the robot manufacturer, resolves to use "cobots" not for their efficiency, but to compensate for the lack of manpower and the lack of space to separate the machines from the workers [1].

However, one could imagine that the autonomy and quality of judgment of the humans should add up to the power and reliability of the robots to create a productive and safe interaction. In fact, the main bottleneck preventing this expected revolution is the required safety guarantee that the robot should provide [2]. Ensuring safety for humans involves many factors, but the most important one is to limit undesired interaction forces [3]. The control of physical interactions is classically divided into three directions [4] : the first one is the avoidance of unwanted contact via space limits or potential fields [5], [6], the second one is the quantitative limitation of energy or velocities to make potential impacts safer [7] and the third one is post-impact control [8]. Since avoiding the contact requires either to predict or to detect the obstacles around, it is out of the scope of this paper, and we focus on setting limits for pre-impact and on the control of external forces. For instance, both these aspects benefit from a key property that is compliance which consists in the passive or active modification of the geometry to reduce the undesired external forces (see Figure 1).

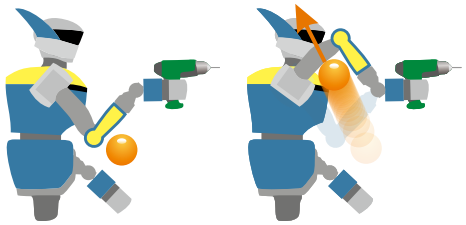

Fig. 1. A sketch to represent the compliance where the robot modifies its geometry to adapt to the external disturbance. For instance, the compliance depicted here is in the null-space of a Cartesian task of a redundant robot, allowing it to continue performing the task despite the adaptation.

Position-controlled robots can deal with some of these constraints. Compliance is possible for robots equipped with variable stiffness actuators [9], but they show lower performance than stiffer ones. For other robots, one can implement schemes that respect velocity limits and torque limits [10]. But the only way these robots may deal with post-contact safety is when this contact occurs with a limb equipped with an external-force sensor.

For this reason, the most successful way of controlling the forces today is to explicitly control the joint torque, since this allows to monitor the forces exerted on the environment at virtually any point. Yet, most solutions for safe interactions rely on robots equipped with joint torque feedback, allowing to produce natural compliance [11], to estimate the external forces when they occur [12] and to control them [13]. However, these sensors are still expensive and fragile. Therefore, several methods were developed to provide similar guarantees without using these sensors to estimate the external forces [14], [15], or to produce compliance [16], [17]. But to our best knowledge, none of these methods has been used with robots having high reduction gears and no torque feedback. Indeed, these robots suffer from a poor backdrivability and a high level of friction [18], and they are usually controlled using high gain-position control. Furthermore, due to their power and their mass, these robots are intrinsically less safe for interacting with humans.

In this paper we present a complete framework for controlling the torque for such robots while having compliance and safety guarantees into account. Several of these features have already been presented in previous publications and will be reminded here, and two novel contributions are presented: (i)

M. Benallegue, R. Cisneros, share the first authorship. M. Benallegue, R. Cisneros, A. Benallegue, A. Tanguy, A. Escande, M. Morisawa, F. Kanehiro are with the CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. A. Benallegue is also with Laboratoire d'Ingénierie des Systèmes de Versailles, France, Email mehdi.benallegue@aist.go.jp, rafael.cisneros@aist.go.jp

a new QP-based anti-windup technique to keep the highest level of performance without generating excessive corrective torques and (ii) the null-space compliance setting allowing the robot to exploit its redundancy and be compliant without degrading the performance of important tasks.

II. Torque control architecture

A. Robot dynamics

Let us consider a rigid multi-body robot with n Degrees of Freedom (DoF) and configuration space $\mathcal{Q}$. We call $\boldsymbol{q} \in \mathcal{Q}$ its configuration, $\boldsymbol{\alpha} \in \mathbb{R}^n$ its configuration velocity and $\dot{\boldsymbol{\alpha}} \in \mathbb{R}^n$ its configuration acceleration. Particularly, if $\mathcal{Q}$ is Euclidian, as it is the focus of this paper and how it will be henceforth considered, then $\boldsymbol{\alpha} = \dot{\boldsymbol{q}}$. However, these results can be extended to the non-Euclidian configuration coordinates such as floating-base orientation [19].

The dynamical model of the robot is written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} + \sum \boldsymbol{J_i}(\boldsymbol{q})^{\boldsymbol{T}} \boldsymbol{F_i}, \quad (1)$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite mass-matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ is a Coriolis matrix for which $\dot{\boldsymbol{M}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is skew-symmetric [20], $\boldsymbol{g} \in \mathbb{R}^n$ is the gravity vector, $\boldsymbol{F_i} \in \mathbb{R}^6$ is the i-th external wrench (force and torque), $\boldsymbol{J_i} \in \mathbb{R}^{6 \times n}$ is the absolute Jacobian of its point of application, and $\boldsymbol{u} \in \mathbb{R}^n$ is a vector of input generalized forces which can include actuated and unactuated DoF (zero entries). From now on, we may omit the functions parameter for vectors and matrices to simplify the notations (e.g. write $\boldsymbol{M}$ and $\boldsymbol{C}$, instead of $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$).

B. Passivity-based integral feedback

An inverse-dynamics control law for (1) is a reference torque $\boldsymbol{u^r}$ calculated as a function of a reference acceleration, $\ddot{\boldsymbol{q}}^{\boldsymbol{r}}$, and a reference vector of external torques, $\boldsymbol{u_e^r}$ as follows

$$\boldsymbol{u^r} = \boldsymbol{M}\ddot{\boldsymbol{q}}^{\boldsymbol{r}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} - \boldsymbol{u_e^r}, \quad (2)$$

Let us add an integral term to (2) to get the torque control law $\boldsymbol{u} = \boldsymbol{u^p}$, defined as

$$\boldsymbol{u^p} = \boldsymbol{u^r} + \boldsymbol{L}\boldsymbol{s} \quad (3)$$

where $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is an integral gain, $\boldsymbol{s} = \dot{\boldsymbol{q}}^{\boldsymbol{r}} - \dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}^{\boldsymbol{r}}(t) = \int_{t_0}^{t} \ddot{\boldsymbol{q}}^{\boldsymbol{r}}(\iota)\, d\iota$.

A passivity-based controller is achieved with

$$\boldsymbol{L} = \boldsymbol{C} + \boldsymbol{K}, \quad (4)$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is any positive definite matrix ($\boldsymbol{K} > 0$).

The control law (3) with (4) achieves exponential stability for $\boldsymbol{s}$ and $\dot{\boldsymbol{s}}$. For the proof see [19].

The gain matrix can be chosen as time-varying $\boldsymbol{K} = \lambda \text{diag}(\boldsymbol{M})$ (see [21]), where $\boldsymbol{M}$ is the mass matrix, $\text{diag}(\cdot)$ returns a diagonal matrix of the diagonal elements of the input matrix and $\lambda > 0$. This gives a weighting factor related to the inertia driven by each joint.

C. Integration in a QP framework

A redundant robot (e.g. a humanoid robot) can simultaneously achieve multiple tasks while satisfying kinematic and dynamic constraints. We use a QP solver to minimize the tracking error for several weighted tasks by computing an optimal reference configuration acceleration, $\ddot{\boldsymbol{q}}^{\boldsymbol{r}}$, and a feasible reference of external forces, $\boldsymbol{u_e^r}$, parametrized[1] by the vector $\boldsymbol{\rho^r}$, subject to linear equality, linear inequality, and bounding constraints. Both $\ddot{\boldsymbol{q}}^{\boldsymbol{r}}$ and $\boldsymbol{u_e^r}$ feed the controller (3) using (2) and (4). The QP problem is written as

$$\begin{bmatrix} \ddot{\boldsymbol{q}}^{\boldsymbol{r}} \\ \boldsymbol{\rho^r} \end{bmatrix} = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{\mathcal{W}}\left(\boldsymbol{A}^{\text{ob}}\boldsymbol{x} - \boldsymbol{b}^{\text{ob}}\right)\right\|^2 + \frac{1}{2}\gamma\left\|\boldsymbol{x}\right\|^2, \quad \text{s.t.} \quad \boldsymbol{A}^{\text{eq}}\boldsymbol{x} = \boldsymbol{b}^{\text{eq}},\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}, \quad (5)$$

where $\boldsymbol{x}$ is the decision variable vector of the optimization problem. Also, $\boldsymbol{\mathcal{W}}$ is a block diagonal matrix made up of weight matrices and γ is a small regularization weight.

Tasks are formulated through the linear system $\left(\boldsymbol{A}^{\text{ob}}, \boldsymbol{b}^{\text{ob}}\right)$, which vertically concatenates the matrices and vectors for k tasks. Constraints are formulated similarly, by vertically concatenating matrices and vectors. Equality constraints are formulated through $(\boldsymbol{A}^{\text{eq}}, \boldsymbol{b}^{\text{eq}})$, inequality constraints through $(\boldsymbol{A}, \boldsymbol{b})$ and boundary constraints through vectors $\boldsymbol{l_b}$ and $\boldsymbol{u_b}$.

Motion-related tasks (in joint or Cartesian space) are specified with acceleration objectives, $\ddot{\boldsymbol{g}}_{\boldsymbol{t}}^{\text{ob}}$, implemented with PD tracking and a feed-forward term. For example, the posture task (in joint space) is defined as $\ddot{\boldsymbol{g}}_{\boldsymbol{t}}^{\text{ob}} = \ddot{\boldsymbol{q}}^{\text{ob}}$, while the position and orientation tasks of a link l in Cartesian space are defined as $\ddot{\boldsymbol{g}}_{\boldsymbol{t}}^{\text{ob}} = \dot{\boldsymbol{v}}_{\boldsymbol{l}}^{\text{ob}}$ and $\ddot{\boldsymbol{g}}_{\boldsymbol{t}}^{\text{ob}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{l}}^{\text{ob}}$, such that

$$\ddot{\boldsymbol{q}}^{\text{ob}} = \boldsymbol{K_{p,q}}\left(\boldsymbol{q^d} - \boldsymbol{q}\right) + \boldsymbol{K_{v,q}}\left(\dot{\boldsymbol{q}}^{\boldsymbol{d}} - \dot{\boldsymbol{q}}\right) + \ddot{\boldsymbol{q}}^{\boldsymbol{d}}, \quad (6)$$

$$\dot{\boldsymbol{v}}_{\boldsymbol{l}}^{\text{ob}} = \boldsymbol{K_p}\left(\boldsymbol{p_l^d} - \boldsymbol{p_l}\right) + \boldsymbol{K_v}\left(\boldsymbol{v_l^d} - \boldsymbol{v_l}\right) + \dot{\boldsymbol{v}}_{\boldsymbol{l}}^{\boldsymbol{d}}, \quad (7)$$

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{l}}^{\text{ob}} = \boldsymbol{K_p}\tilde{\boldsymbol{\Omega}} + \boldsymbol{K_v}\left(\boldsymbol{\omega_l^d} - \boldsymbol{\omega}_l\right) + \dot{\boldsymbol{\omega}}_{\boldsymbol{l}}^{\boldsymbol{d}}, \quad (8)$$

where $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are diagonal matrices of PD gains and $\tilde{\boldsymbol{\Omega}} = \boldsymbol{S^{-1}}(\log\left\{\boldsymbol{R_l^d}\boldsymbol{R_l^T}\right\})$ calculates the error vector in orientation. The super-script d stands for desired values, terms without subscript indicate current values and $\boldsymbol{S^{-1}}(\cdot) : \mathbb{R}^{3\times3} \to \mathbb{R}^3$ is the inverse of the skew-symmetric operator.

Then, for task t, $\boldsymbol{A_t^{\text{ob}}}$ and $\boldsymbol{b_t^{\text{ob}}}$ are given by

$$\boldsymbol{A_t^{\text{ob}}} = \boldsymbol{J_{g,t}}(\boldsymbol{q}), \qquad \boldsymbol{b_t^{\text{ob}}} = \ddot{\boldsymbol{g}}_{\boldsymbol{t}}^{\text{ob}} - \dot{\boldsymbol{J}}_{\boldsymbol{g,t}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}, \quad (9)$$

where $\boldsymbol{J_{g,t}}(\boldsymbol{q})$ and $\dot{\boldsymbol{J}}_{\boldsymbol{g,t}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the task Jacobian and its time derivative.

Furthermore, it is possible to realize task-space force control by adding wrench and admittance-like tasks as it will be described on Section III-E, or even balance-related tasks in case of mechanisms with floating base (e.g. humanoid robots), as detailed in [21].

III. Safety features

A. The cost of integral convergence

It is important to remind that the guarantee of convergence of $\boldsymbol{s}$ provided by the control law (3) is actually stronger than

[1] External forces can be represented as a vector of non-negative coefficients, $\boldsymbol{\rho}$, that constrain these forces to be inside of pyramidal approximations of the corresponding friction cones. See [22] or [21] for more details.

a simple convergence of velocity error to zero. In fact, since $\ddot{\boldsymbol{q}}^r$ results from a closed-loop control law, the convergence of $\boldsymbol{s}$ amounts at a convergence of the integral of the joint position error [19].

This is usually a desired feature since it allows to overcome modeling errors and biases on the actual generated joint torque. These problems are especially important in the case of robots with high gear ratio and no joint-torque feedback. However, having an integral convergence is also potentially dangerous when the modeling error cannot be overcome with stronger forces or when a contact with humans is involved. Therefore we need to mitigate this control with a collection of solutions that we present hereinafter.

B. QP-based Anti-windup

An anti-windup technique allows to guarantee that integral terms do not go beyond safety limits [23]. In our case, the torque additional term, constituted with $\boldsymbol{Ls} = (\boldsymbol{C}+\boldsymbol{K})\boldsymbol{s}$, can grow arbitrarily big and might require to limit its value. Since $\boldsymbol{Ls}$ is a vector, we could consider using traditional techniques on each of its components, and it would successfully limit the term to any desired value. However such a naive approach would have an undesired side effect.

Indeed, one condition of the convergence of the control law (3) is the positive definiteness of $\boldsymbol{K}$, in other terms the dot product between the vectors $(\boldsymbol{L}-\boldsymbol{C})\boldsymbol{s}$ and $\boldsymbol{s}$ should be positive. It is clear that altering the vector components independently each from other may change its direction and thus violate this condition.

Nevertheless, we can notice that when this term is winding up, the part $\boldsymbol{Ks}$ constitutes most of it. This is because $(\boldsymbol{C}+\boldsymbol{K})\boldsymbol{s}$ represents only the inertial component of the joint velocity error energy. Therefore, if we simply reduce the eigenvalues of $\boldsymbol{K}$ while keeping them positive, this term would decrease. This is why we propose to limit only $\boldsymbol{u^K} = \boldsymbol{Ks}$ by indirectly reducing the eigenvalues of $\boldsymbol{K}$.

For instance, let us say that we want to keep $-\boldsymbol{u}^m < \boldsymbol{u^K} < \boldsymbol{u}^m$. One first approach would be simply to multiply this vector by the factor corresponding to the biggest relative violation of the torque maximum value [21]. In other words we multiply $\boldsymbol{u^K}$ by λ_w defined as

$$\lambda_w = \min\left(1, \min_i \frac{(\boldsymbol{u}^m)_i}{|(\boldsymbol{u^K})_i|}\right). \tag{10}$$

where $(\boldsymbol{u})_i$ is the i-th component of $\boldsymbol{u}$. This solution guarantees to respect the proof of convergence and to respect these boundary values. However, this would degrade the whole performances of the robot as soon as the torque of any joint is reaching its maximum value, and in the case of robots performing multiple tasks, this effect is uselessly constraining for the tasks that do not involve this joint.

Instead, we propose here the solution to replace $\boldsymbol{u^K}$ with $\tilde{\boldsymbol{u}}^{\boldsymbol{K}}$ resulting from the following QP optimization

$$\begin{aligned} \tilde{\boldsymbol{u}}^{\boldsymbol{K}} = \arg\min_{\boldsymbol{x}} \ & \|\boldsymbol{x} - \boldsymbol{u^K}\|^2 \\ \text{s.t.} \ & -\boldsymbol{u}^m \leq \boldsymbol{x} \leq \boldsymbol{u}^m \\ & \boldsymbol{s}^T\boldsymbol{x} \geq \lambda_w \boldsymbol{s}^T\boldsymbol{u^K} \end{aligned} \tag{11}$$

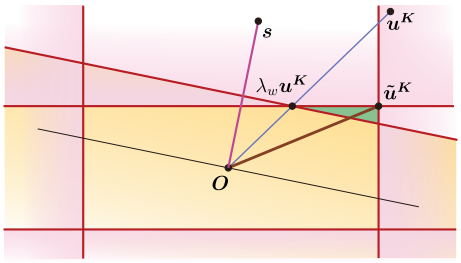

Fig. 2. A simplified diagram of the QP-based anti windup. The pink areas are violating the bounds on the torque and the orange area is violating the positive definiteness of the solution. The feasible area is then the green triangle where the solution $\tilde{\boldsymbol{u}}^{\boldsymbol{K}}$ is the closest possible to its unconstrained reference $\boldsymbol{u^K}$.

This means that we need to get as close as possible to the original value of $\boldsymbol{u^K}$, within the torque boundaries, but that do not degrade the positive definiteness of the new term compared to the solution $\lambda_w \boldsymbol{u^K}$ from our previous approach. We are guaranteed that this QP has a solution and will never fail since $\lambda_w \boldsymbol{u^K}$ satisfies all the constraints. A 2D example is provided in Figure 2.

C. Leaky integration

The main reason for the winding up of the variable $\boldsymbol{u^K}$ is that $\boldsymbol{s}$ may grow unbounded. But this mainly happens when the robot hits an obstacle or a too big discrepancy between the model and the measured state. This kind of modeling error is not the target of the integral term, which is designed to account for (i) friction and (ii) biases between the command torque and the real one, as well as (iii) errors in the mass distribution model. For the kind of robots we are studying, the most important component of friction is static friction which acts only at the start of a motion. The other kinds of errors depend on the state of the robot: kinetic friction depends on the velocity and the effect mass distribution error depends mostly on the configuration. This means that all of these disturbances evolve with time and should not be integrated indefinitely. However, a regular integrator has permanent memory and a perturbation needs to be compensated for even if it happened minutes before.

This is the main motivation for the implementation of the leaky integration $\boldsymbol{s}^\theta$ for $\boldsymbol{s}$ which can be written in the following way

$$\dot{\boldsymbol{s}}^\theta = -\theta \boldsymbol{s}^\theta + \dot{\boldsymbol{s}} \tag{12}$$

and since we do not have a direct measurement for $\dot{\boldsymbol{s}}$ but only discrete time measurements of $\boldsymbol{s}_k$, this is actually implemented as

$$\boldsymbol{s}_k^\theta = \exp\left(-\theta\Delta_t\right)\boldsymbol{s}_{k-1}^\theta + (\boldsymbol{s}_k - \boldsymbol{s}_{k-1}) \tag{13}$$

This filter allows to set up a tunable memory where $\theta = \frac{1}{t_l}$ defines the time t_l such that $1 - \exp(-1) = 63\%$ of the integral is forgotten.

For instance, we want to focus on specific kinds of disturbances described above and we isolate short-term perturbations (static friction) and state-dependent ones. These define two time spans, that can be targeted using two different

values of θ: a slow one and a fast one. Therefore, we mix these two leaky integrations to build the selected solution

$$\tilde{\boldsymbol{s}} = \mu \boldsymbol{s}^{\theta_f} + (1-\mu)\, \boldsymbol{s}^{\theta_s} \tag{14}$$

where θ_f and θ_s are respectively the fast forgetting and the slow forgetting values of θ, and $0 \leq \mu \leq 1$ is a weighing factor between them.

With this setting, the short disturbances are quickly compensated for and forgotten by the fast filter, and the slower disturbances are mostly dealt with using the slow filter.

Note that by developing (14) we can sum the filters to obtain a second order system, but as we will see in the next section, we prefer to keep it in this form to allow some combinations enabling specific desired behaviors.

D. Null-space compliance

This kind of behavior allows the robot to take profit from its redundancy and adapt its geometry to external forces without interfering with an important task. This task is usually a Cartesian task requiring less degrees of freedom than those available to accomplish it. The remaining degrees of freedom are called null space of the task and involve usually all the joints.

We propose a modification of the control law proposed in (3) to provide a strong guarantee on the convergence of the task even when being compliant in the null space.

Let's define the task function $\boldsymbol{T}(\boldsymbol{q}) \in \mathbb{R}^m$ $(m < n)$, the time-derivatives of this function with regard to the configuration are computed thanks to $\boldsymbol{J}(\boldsymbol{q})$ the task Jacobian with regard to the configuration, that we consider full rank. This gives the following identities: $\dot{\boldsymbol{T}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{T}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$. For simplicity, we will simply write $\boldsymbol{J}$ and $\dot{\boldsymbol{J}}$. The reference configuration acceleration $\ddot{\boldsymbol{q}}^r$ which depends on the robot state $\left(\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T\right)^T$ defines the reference acceleration of the task $\ddot{\boldsymbol{T}}^r = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}^r + \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$. This reference is usually defined such that $\boldsymbol{T}$ converges to a desired reference $\boldsymbol{T}^*$ (e.g. see (7) and (8)), but in this section we do not need to enter into these details. What is important is the error that occurs in this task when the reference joint acceleration $\ddot{\boldsymbol{q}}^r$ is not respected. Let's define this acceleration error

$$\ddot{\boldsymbol{e}}_T = \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{J}\ddot{\boldsymbol{q}}^r - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \boldsymbol{J}\ddot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{s}} \tag{15}$$

Let's define the variables $\boldsymbol{a}$ and $\dot{\boldsymbol{q}}^a$ as follows

$$\boldsymbol{a} = \boldsymbol{s} - \boldsymbol{J}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) dt \tag{16}$$

$$\dot{\boldsymbol{q}}^a = \dot{\boldsymbol{q}}^r - \boldsymbol{J}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) dt \tag{17}$$

$$\text{giving that} \quad \dot{\boldsymbol{a}} = \ddot{\boldsymbol{q}}^r - \ddot{\boldsymbol{q}} - \boldsymbol{J}^+\dot{\boldsymbol{J}}\boldsymbol{s} - \dot{\boldsymbol{J}}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) dt \tag{18}$$

$$\ddot{\boldsymbol{q}}^a = \ddot{\boldsymbol{q}}^r - \boldsymbol{J}^+\dot{\boldsymbol{J}}\boldsymbol{s} - \dot{\boldsymbol{J}}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) dt \tag{19}$$

where $\boldsymbol{J}^+$ is a $(m \times n)$ generalized inverse such as the pseudo-inverse or the dynamically consistent generalized inverse [24]. For simplicity, in the next developments we will omit the integration variable.

The proposed control law is given by

$$\boldsymbol{u}^a = \boldsymbol{M}\ddot{\boldsymbol{q}}^a + \boldsymbol{C}\dot{\boldsymbol{q}}^a + \boldsymbol{g} - \sum_i \boldsymbol{J}_i^T \boldsymbol{F}_i + \boldsymbol{K}\boldsymbol{a} \tag{20}$$

with $\boldsymbol{K}$ a positive definite gain matrix.

Equation (20) can be rewritten as

$$\boldsymbol{u}^a = \boldsymbol{u}^r + \boldsymbol{C}\boldsymbol{a} - \boldsymbol{M}\boldsymbol{J}^+\dot{\boldsymbol{J}}\boldsymbol{s} - \boldsymbol{M}\dot{\boldsymbol{J}}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) + \boldsymbol{K}\boldsymbol{a} \tag{21}$$

with $\boldsymbol{u}^r = \boldsymbol{M}\ddot{\boldsymbol{q}}^r + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} - \sum_i \boldsymbol{J}_i^T \boldsymbol{F}_i$.

The closed-loop dynamics obtained by setting $\boldsymbol{u} = \boldsymbol{u}^a$ gives

$$\boldsymbol{M}\left(\ddot{\boldsymbol{q}}^r - \ddot{\boldsymbol{q}} - \boldsymbol{J}^+\dot{\boldsymbol{J}}\boldsymbol{s} - \dot{\boldsymbol{J}}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right)\right) = -\boldsymbol{C}\boldsymbol{a} - \boldsymbol{K}\boldsymbol{a} \tag{22}$$

By using equation (18), we get

$$\boldsymbol{M}\dot{\boldsymbol{a}} = -\left(\boldsymbol{C} + \boldsymbol{K}\right)\boldsymbol{a} \tag{23}$$

Theorem 1: Consider the robot dynamics (1) with the control law $\boldsymbol{u} = \boldsymbol{u}^a$ defined by (20)-(16)-(17)-(19),] and resulting in the closed loop error dynamics given by (23). Then exponential convergence of the variable a to zero is guaranteed, leading to the convergence of $\int \ddot{\boldsymbol{e}}_T$ to zero.

Proof: Let's take the following Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a} \tag{24}$$

Its time derivative is obtained using the error dynamics (23)

$$\dot{V} = \frac{1}{2}\boldsymbol{a}^T\dot{\boldsymbol{M}}\boldsymbol{a} + \boldsymbol{a}^T\boldsymbol{M}\dot{\boldsymbol{a}} = -\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a}$$

where we used the fact that $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew symmetric. This leads to the exponential convergence of $\boldsymbol{a}$ to zero which leads to the convergence of $\boldsymbol{J}\boldsymbol{a}$ to zero.

By using the fact that $\boldsymbol{J}\boldsymbol{J}^+ = \boldsymbol{I}$ we notice that

$$\boldsymbol{J}\boldsymbol{a} = \boldsymbol{J}\boldsymbol{s} - \boldsymbol{J}\boldsymbol{J}^+ \int \left(\dot{\boldsymbol{J}}\boldsymbol{s}\right) = \int \ddot{\boldsymbol{e}}_T \tag{25}$$

Which concludes the proof. ■

This convergence guarantee is of the same kind as the convergence of $\boldsymbol{s}$ in the sense that it provides not only the velocity convergence to zero, but if it is replaced in the closed loop control of $\ddot{\boldsymbol{T}}^r$ it provides the convergence of the integral of the task error. So it will guarantee a better tracking of the reference task dynamics and has very little effect on the null-space dynamics through the derivative-related feedforward terms at the end of (21).

This scheme is comparable to the disturbance observer solution presented in [17] for their null space compliance, where an integrator is used to estimate the torque disturbance from the error in the task space, and is compensated for. But their tests were preformed on a lightweight robot with torque feedback, which faces lower levels of modeling errors and is able to work with lower integral terms.

Finally, the robot has a better null-space compliance if the term $\boldsymbol{Ka}$ in (20) is set in the following way

$$(1-\mu)\boldsymbol{J^T J^{+T} K J^{+} J a^{\theta_s}} + \mu \boldsymbol{K a^{\theta_f}} \tag{26}$$

where $\boldsymbol{a^{\theta_s}}$ and $\boldsymbol{a^{\theta_f}}$ are respectively the result of slow-forgetting and fast-forgetting leaky integration. This scheme allows to keep an overall positive definite gain matrix but leave only the task space for the slow correction to not downgrade the performance of the task.

E. Admittance control and additional features

This control framework offers other safety features that monitor some interaction forces and ensure the feasibility of the motion.

The most prominent of these features is the admittance control where we take profit from the external force sensors to control forces. These forces need to be calculated within the QP to guarantee their feasibility. The contact forces and moments (wrenches), parametrized by $\boldsymbol{\rho_r}$, can be realized if the reference acceleration $\boldsymbol{\ddot{q}_r}$ is adequately tracked provided a perfect model of the environment. If not, the actual forces and moments will differ from the calculated ones. Therefore we need a control loop to compensate for these discrepancy.

With the QP it is straightforward to specify tasks in Cartesian space, and therefore to implement a position-based force control. To do that consistently with the forces calculated by the QP it is necessary to implement it in two steps (see [25] for more details):

(a) First, we need to "drive" the QP to exert a desired wrench with a certain link, by using a *wrench task* in the space of the wrenches to minimize $\frac{1}{2}\left\| \begin{bmatrix} \mathbf{0}_{\text{ini}} & \boldsymbol{W_l} & \mathbf{0}_{\text{fin}} \end{bmatrix} \boldsymbol{\rho^r} - \boldsymbol{F_a^d} \right\|^2$. where $\boldsymbol{W_l}$ represents a wrench matrix that "decodes" the portion of $\boldsymbol{\rho}$ corresponding to link l into a wrench, $\mathbf{0}_{\text{ini}}$ and $\mathbf{0}_{\text{fin}}$ are leading and trailing zero matrices, and $\boldsymbol{F^d} \in \mathbb{R}^6$ is the desired wrench.

(b) Then, we need to ensure the application of the optimal wrench via force / moment control, on a link for which the exerted wrench can either be measured (using a 6-axes force sensor) or estimated. The way to do that is by implementing an *admittance task* that aims to indirectly regulate the wrench on a link by modifying its position reference. This task can be implemented through (9) by specifying the following acceleration objective:

$$\boldsymbol{\ddot{g}_t^{\text{ob}}} = \begin{bmatrix} \boldsymbol{\dot{v}_l^{\text{ob}}} \\ \boldsymbol{\dot{\omega}_l^{\text{ob}}} \end{bmatrix} = -\boldsymbol{K_p}\left(\boldsymbol{F_l^r} - \boldsymbol{F_l}\right) - \boldsymbol{K_v}\left(\boldsymbol{\dot{F}_l^r} - \boldsymbol{\dot{F}_l}\right), \tag{27}$$

where $\boldsymbol{F_l^r}$ and $\boldsymbol{F_l}$ are the reference and actual wrenches, while $\boldsymbol{\dot{F}_l^r}$ and $\boldsymbol{\dot{F}_l}$ correspond to their time derivatives (calculated by finite differences). It is worth to mention that $\boldsymbol{F_l^r}$ is calculated from the $\boldsymbol{\rho_r}$ calculated by the QP in the previous iteration; that is, there is one time step of delay.

This admittance task shares the same task Jacobian as the one of the position and orientation tasks of the corresponding link, such that their effect gets combined to achieve hybrid or parallel control by setting the stiffness of the pose task to zero in the required direction(s). Notice that the damping of the pose task should not be zero, to avoid potential instabilities.

If there are two or more link simultaneously controlled in force it is better to use an improvement of (27) that projects the internal forces into the null-space of the force distribution matrix, as described in detail in [21]; however, as it is not required to illustrate the contributions previously made by this paper, we will not describe it.

Other important features consist in safety and feasibility constraints. To ensure that they are effectively respected, the QP must be aware of the integral term of (3) and has to take it into account by explicitly regarding it as an "artificial external generalized force" within the expressions describing the constraints [19].

Another important safety constraint is torque limits. In our framework these are guaranteed regardless of the value of integral and additional terms. This is an important feature since it allows to limit the kinetic energy or the forces in case of impact.

Feasibility constraints are also necessary for safety, since they ensure that the predicted dynamics will be respected by the robot. Among these constraints, we can cite the feasibility of the unactuated DoF dynamics (if there is one), the respect of joint range and speed limits, collision avoidance constraints, and contact unilaterality / friction constraint ($\boldsymbol{\rho_r} > 0$) [25].

IV. Experimental validation

A. Experimental setup

To validate the above-described safety features, we tested them using our humanoid robot HRP-5P. It has 37-dof, a height of 1.83 m, a weight of 101 kg, and powerful joints driven by high reduction Harmonic Drive gears (200 to 300) [26]. The motor drivers accept torque commands, and can give as a feedback the torque generated at the motor level (calculated from the current), but there are no torque sensors at the joint level that can give information on the magnitude of the joint friction. The arm has a 6D force/torque sensor on the end-effector.

The QP-based torque control is developped in the open source `mc_rtc` framework[2], which was modified to implement the safety features, while running over `OpenRTM-aist`[3]. The QP-based anti-windup protection was implemented using the open source `jrl-qp` solver[4].

The control system of HRP-5P is configured to start in position control, and to do that the default behavior of the QP solver of `mc_rtc` is to compute the reference joint angles by double integrating the reference joint acceleration; that is, $\boldsymbol{q^r} = \int\int \boldsymbol{\ddot{q}^r}$. This does not work in closed-loop, but in open loop; that is, by using the result of the double integration as the current state for the next iteration [27].

This means that switching from position to torque control is not straightforward. There are two considerations that have to be taken into account:

[2] `https://jrl-umi3218.github.io/mc_rtc`
[3] `https://www.openrtm.org/openrtm`
[4] `https://github.com/jrl-umi3218/jrl-qp`

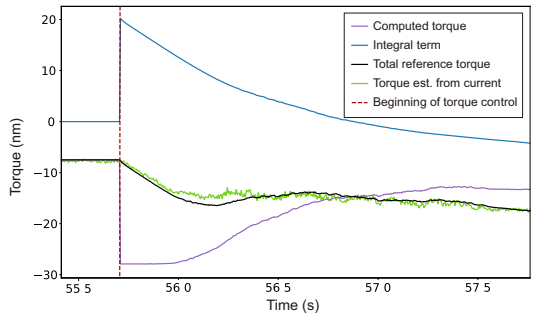

Fig. 3. Smooth transition from position to torque control.

TABLE I

DEFAULT TASK PARAMETER VALUES.

Task	Param.	Value	Task	Param.	Value
q	PD	0, 5	poRH	PD	900, 50
	weight	10		weight	10000
wrRH	weight	10	adRH	f-PD	5e-3, 1e-7
	f-des	(0, 0, 0)		n-PD	5e-3, 1e-7
	n-des	(0, 0, 0)		weight	10000

(1) The transition has to be done in two consecutive time steps: (i) First, the current reference joint angle has to be "captured" with a *state holder* at the input of the PD controller, while the QP is given the current joint configuration of the robot. By doing this, the QP will generate at the next step the right torque reference. (ii) Then, the torque reference can be given to the robot.
(2) The integral term has to be initialized to the difference between these two torques, such that the torque reference given to the motors would be continuous.

The above-mentioned considerations were taken into account for the hereinafter described experiments. Figure 3 shows the elbow joint at the moment of the transition and it can be assessed that the reference torque is continuous. If that had not been done, the reference torque would have changed to the computed torque (purple signal). However, as the integral term (blue signal) was initialized to the difference, the sum of both (black signal) was continuous. This behavior can also be observed in the accompanying video.

As for the experiments described next, the QP solver was specifically instructed to consider only the joints of the right arm and ignore the remaining, as well as the floating base. The tasks that were used are mentioned (and nicknamed) as follows. The posture (configuration) task (`q`), the pose task of the Right Hand (`poRH`), the wrench task for it (`wrRH`) and the corresponding admittance task (`adRH`). Also, by default, the considered constraints are the torque limits, as well as joint range and speed limits. The default PD gains and weights are shown in Table I. Notice that for the Admittance Tasks the PD gains of the force and moment components are indicated by `f-` and `n-`, respectively.

Regarding the torque control, the parameter values were $\lambda = 2$, $\boldsymbol{\tau}^m = 0.1\boldsymbol{\tau}^l$ where $\boldsymbol{\tau}^l$ is the joint torque limit vector, $\theta_s = 0.1s^{-1}$, $\theta_f = 100s^{-1}$ and $\mu = 0.5$.

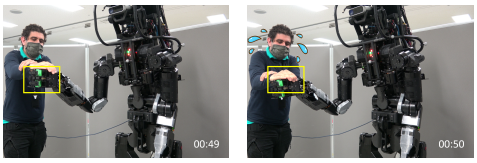

Fig. 4. An illustration of the test of anti-windup. The hand of the robot is pushed for several seconds then released.

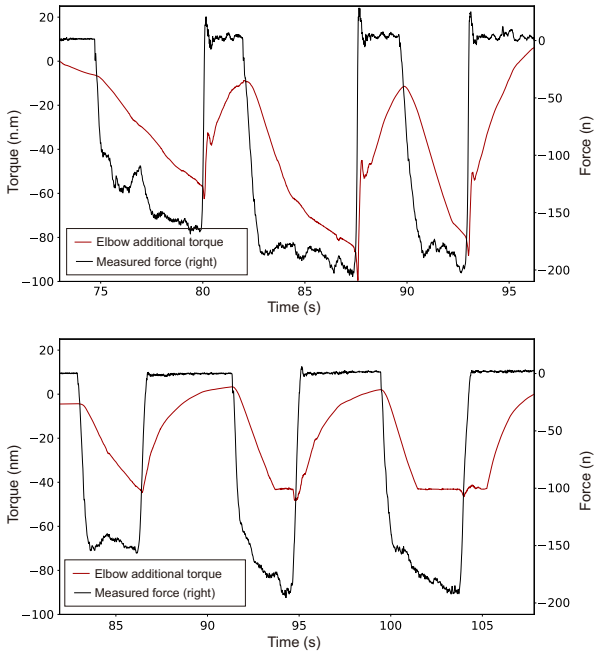

Fig. 5. **Plots for the controller without anti-windup on the top and with it on the bottom.** In red is the additional torque on the elbow joint and in black it is measured force. Since they are on different scales, the force axis is on the right of the plot.

B. Anti-windup

To test the anti-windup we compare the controller with and without it. The experiment consists in applying a disturbance force on the hand of the robot and waiting for several seconds to make the integral term build up until reaching the set limit (see Figure 4 for illustration).

We can see in Figure 5 the effect of this disturbance force making the additional torque grow as a leaky integrator. On the top we see that without anti windup, this grow is unbounded and causes a higher oscillation when released (visible on the force plot). On the bottom we see the anti-windup saturating the additional term signal and no visible oscillation at the release of the force. It is interesting to notice that the additional term goes beyond the limit when the force is released, this is due to the fact that only the linear gain part of the additional terms is saturated, and not the inertial ones, and these were more important during the first instants after the release due to high velocities.

C. Null-space compliance control for a static task

To test the null-space compliance, we simply asked the robot to keep a Cartesian position and orientation of the hand

and we test both the stiffness of this task and the compliance in its null-space, mostly by pushing on the elbow joint.

Three scenarios are tested: the first one is the default control parameters without the null-space compliance. We see in Figure 7 snapshots of this experiment. The robot hardly moves in any joint. The task is respected with big stiffness. Regarding the null-space, we can move the elbow by pushing on it but with a lot of resistance. This is due to the absence of stiffness in the posture task, so if we manage to change the posture in the null-space there will be no torque to bring it back to a reference (top of Figure 6).

The second scenario differs by the activation of the null-space compliance control law from (21). We see in Figure 8 snapshots of this experiment. The Cartesian task was also stiff despite the lower gains. The main difference in behavior is that it was much easier to move the robot in the null-space of the task. There were 3 degrees of redundancy, two of them were movable, but the third involved many joints and has a very short leverage. Since this compliance was passive and involves several joints with high friction, it still required some force to move it, but it is virtually impossible to obtain a better behavior from passive compliance on this robot. The bottom of Figure 6 shows the task position error together with a joint angle that shows its trajectory in the task null-space.

The third experiment was a combination of null-space compliance and active compliance produced by admittance control. When the measured force at the wrist was below a threshold, the reference Cartesian position was maintained. When it was above, compliance control was activated and allowed us to modify the position of the end-effector, this new position became the new reference when the end-effector was released, and null-space compliance was again active. We call this behaviour "dual compliance". We see in Figure 9 snapshots of this experiment. This demonstration is interesting because it mixes two different kinds of compliance, and it is only possible when the robot is able to distinguish between a contact occurring on the end-effector that would trigger the admittance and contacts occurring on other links that are only used in the task null space.

D. Null-space compliance control for a time-varying task

In this last experiment, the robot was asked to produce a 6D trajectory with its end-effector. This trajectory was to keep constant orientation and x-axis position and to draw several times a 40cm diameter closed shape in the y-z plane. However we have placed rigid poles that the robot is not aware about, and on which the arm of the robot collides during the task execution (see Figure 10). This situation may occur in unstructured or moving environments, and it usually leads to a failure of the robot if not a threat on the environment and surrounding humans. Therefore, in this experiment we only tested the null-space compliant controller.

The experiment was safely performed despite the high force impact. On Figure 11 we see the quality of the tracking in x and y axes. The disturbance is visible in the error plots, but we see that it is corrected for and it remains under 2cm.

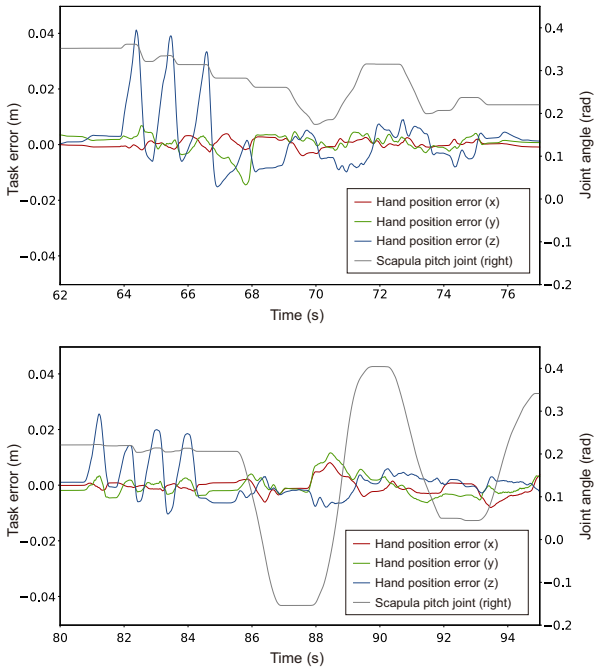

Fig. 6. **Plots showing the Cartesian task errors with one witness joint angle, without null-space compliance on the top and with it on the bottom**. Since the angles and error signals are on different scales, the axis for the joint angle is on the right. In the beginning we push on the hand to test the stiffness of the task and then attempt to move in the task null-space.

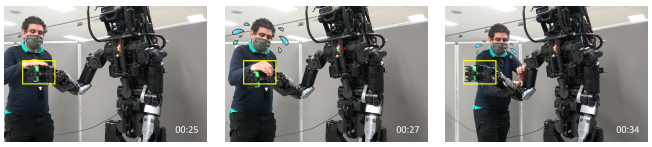

Fig. 7. Testing the control without the proposed null-space compliance, the hand position is stiff and it is difficult to move in the null-space.

V. Conclusion

In conclusion we have shown a torque control framework for high reduction gear robots without torque feedback. This framework is able to exploit safety features implementd originally for position control, such as admittance and feasibility constraints, and where we can develop additional guarantees. The first guarantee is to have a limited integral term, thanks to a well grounded QP-based anti-windup and a system of filters to target specific disturbances. The second guarantee is to converge to a given task, and to be able to limit interference with the null-space of this task.

It is also worth to note that this paper is the first work showing the real robot HRP-5P moving using the torque control scheme. Furthermore, we were able to validate the effect of the new safety and compliance features despite the high level of friction and the low quality of the feedback.

It seems clear to us that the next improvement shall be to pre-compensate for the friction in order to produce active compliance and smoother motions [18].

Fig. 8. Testing the control with null-space compliance. The hand position is still stiff but now it is much easier to move the robot in the null space of this task (2 DoF tested.)

Fig. 9. Testing the dual compliance, when moving on the arm, only the null-space of the task is compliant, therefore the hand remains in the same position, but when interacting with the hand itself the admittance control is triggered and the end effector can be moved freely.

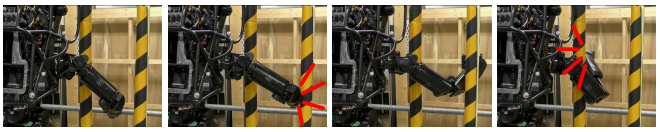

Fig. 10. Testing the null-space compliance during a trajectory tracking task with unmodeled obstacles. The red lines show the collisions.

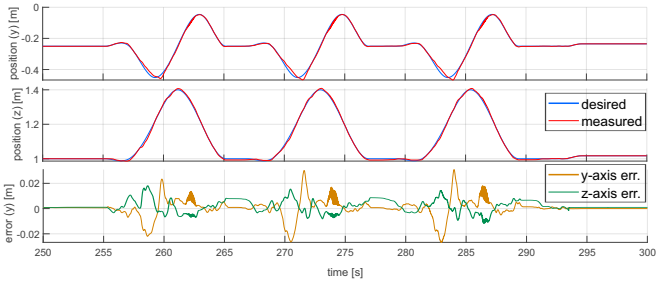

Fig. 11. The result of the trajectory tracking in y and z axes. On the top a comparison between the references and the measured trajectories and in the bottom the corresponding tracking errors.

REFERENCES

[1] Universal robots. Alpha corporation gets solution of automation and improved productivity to the existing production line with flexible ur cobots. https://www.universal-robots.com/case-stories/alpha-corporation/.

[2] V. Murashov, F. Hearl, and J. Howard. Working safely with robot workers: Recommendations for the new workplace. *Journal of occupational and environmental hygiene*, 13(3):D61–D71, 2016.

[3] V. Villani, F. Pini, F. Leali, and C. Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248 – 266, 2018.

[4] P. A Lasota, T. Fong, and J. A. *et al* Shah. *A survey of methods for safe human-robot interaction*. Now Publishers, 2017.

[5] P. A Lasota, G. F. Rossano, and J. A. Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 339–344. IEEE, 2014.

[6] F. Flacco, T. Kröger, A. De Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *2012 IEEE International Conference on Robotics and Automation*, pages 338–345. IEEE, 2012.

[7] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer. A truly safely moving robot has to know what injury it may cause. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5406–5413. IEEE, 2012.

[8] J. Heinzmann and A. Zelinsky. Quantitative safety guarantees for physical human-robot interaction. *The International Journal of Robotics Research*, 22(7-8):479–504, 2003.

[9] M. Lendermann, B. R. P. Singh, F. Stuhlenmiller, P. Beckerle, S. Rinderknecht, and P. V. Manivannan. Comparison of passivity based impedance controllers without torque-feedback for variable stiffness actuators. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1126–1131, 2015.

[10] M. Djeha, A. Tanguy, and A. Kheddar. Adaptive-gains enforcing constraints in closed-loop qp control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2020.

[11] A. Calanca, R. Muradore, and P. Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME Transactions on Mechatronics*, 21(2):613–624, 2016.

[12] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630. IEEE, 2006.

[13] Bojan Nemec, Nejc Likar, Andrej Gams, and Aleš Ude. Human robot cooperation with compliance adaptation along the motion trajectory. *Autonomous robots*, 42(5):1023–1035, 2018.

[14] M. Geravand, F. Flacco, and A. De Luca. Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. *2013 IEEE ICRA*, pages 4000–4007, 2013.

[15] M. Benallegue, P. Gergondet, H. Audren, A. Mifsud, M. Morisawa, F. Lamiraux, A. Kheddar, and F. Kanehiro. Model-Based External Force / Moment Estimation For Humanoid Robots With No Torque Measurement. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[16] Liang Han, Wenfu Xu, Bing Li, and Peng Kang. Collision detection and coordinated compliance control for a dual-arm robot without force/torque sensing based on momentum observer. *IEEE/ASME Transactions on Mechatronics*, 24(5):2261–2272, 2019.

[17] Hamid Sadeghian, Luigi Villani, Mehdi Keshmiri, and Bruno Siciliano. Task-space control of robot manipulators with null-space compliance. *IEEE Transactions on Robotics*, 30(2):493–506, 2013.

[18] R. Cisneros, M. Benallegue, R. Kikuuwe, M. Morisawa, and F. Kanehiro. Reliable chattering-free simulation of friction torque in joints presenting high stiction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 20200*. IEEE, 2020.

[19] 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 International Conference on Intelligent Robots and Systems*, 2018.

[20] M. Bjerkeng and K.Y. Pettersen. A new Coriolis matrix factorization. In *IEEE Internacional Conference on Robotics and Automation*, 2012.

[21] R. Cisneros-Limon, M. Morisawa, M. Benallegue, A. Escande, and F. Kanehiro. An inverse dynamics-based multi-contact locomotion control framework without joint torque feedback. *Advanced Robotics*, 34(21-22), 2020.

[22] M.A. Hopkins, D.W. Hong, and A. Leonessa. Compliant Locomotion Using Whole-Body Control and Divergent Component of Motion Tracking. In *IEEE Internacional Conference on Robotics and Automation*, 2015.

[23] Sergio Galeani, Sophie Tarbouriech, Matthew Turner, and Luca Zaccarian. A tutorial on modern anti-windup design. *European Journal of Control*, 15(3):418–440, 2009.

[24] Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The international journal of robotics research*, 14(1):19–36, 1995.

[25] R. Cisneros, M. Benallegue, M. Morisawa, and F. Kanehiro. QP-based task-space hybrid / parallel control for multi-contact motion in a torque-controlled humanoid robot. In *IEEE-RAS Humanoids*, 2019.

[26] K. Kaneko, H. Kaminaga, T. Sakaguchi, S. Kajita, M. Morisawa, I. Kumagai, and F. Kanehiro. Humanoid Robot HRP-5P: an Electrically Actuated Humanoid Robot with High Power and Wide Range Joints. *IEEE Robotics and Automation Letters (RA-L)*, 4(2), 2019.

[27] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro. Multi-contact vertical ladder climbing by an HRP-2 humanoid. *Autonomous Robots*, 40(3):561–580, March 2016.