Safety Methods for Cartesian Control of Redundant Robotic Arms

Juan Antonio Delgado-Guerrero, Adrià Colomé, Sergi Foix, and Carme Torras
Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Spain
{jdelgado,acolome,sfoix,torras}@iri.upc.edu

***Abstract*—Over the last years, the interest in robot safety has increased, due to the improvements in applications such as assistive robotics, co-manipulation or manufacturing cobots. In these scenarios that include physical interactions between humans and robots, it is of upmost importance to have both passive and active safety layers that prevent robotic arms to behave in an unsafe manner. Ultimately, the safety measures that can be applied might depend on the sensors installed on a robot. Robots with joint torque sensors often base their safety in reactive behaviors when a contact is detected. However, robots should be able to also have safety layers that do not depend on detected contacts. In this paper, we show several additions to robotic Cartesian controllers that can significantly improve safety in human-robot physical interaction for robots with joint position encoders only.**

I. Introduction

In recent years, approaches to have shared spaces between humans and robots have become more popular because of the increasing needs and applications in collaborative and assistive robotics [1]. In the first place, collaborative robots are designed to work together with people in shared spaces. This requires these robots to be very adaptive and respond fast to human actions, which are often hard to predict. The difficulty in predicting future human behaviors makes prediction unreliable for a safe human-robot physical interaction. Therefore, many robotics platforms use joint torque sensors and implement safety layers based on unforeseen contacts. However, how to improve safety in these interactions when we do not have a force/torque sensor feedback? In these scenarios, approaches like feed-forward controllers, that add a model of the robot's inverse dynamics to the control action, allow for lower gains that are intrinsically safer. This then facilitates the implementation of Variable Impedance Control (VIC), which can adjust the robot stiffness and damping dynamically, increasing safety and effectiveness in collaborative tasks according to specific situations. Thus, by modulating their impedance, robots can become compliant and behave gently when necessary, and adapt to be more precise and robust when handling delicate tasks.

Furthermore, VIC has proved to be very useful when it comes to applications such as assistive robotics [2]–[4], which are generally addressed to aid the elderly and disabled, or telemanipulation [5]. In such cases, this kind of control allows for personalizing the degree of assistance in terms of support and resistance, tailoring them to the specific needs and conditions of the users.

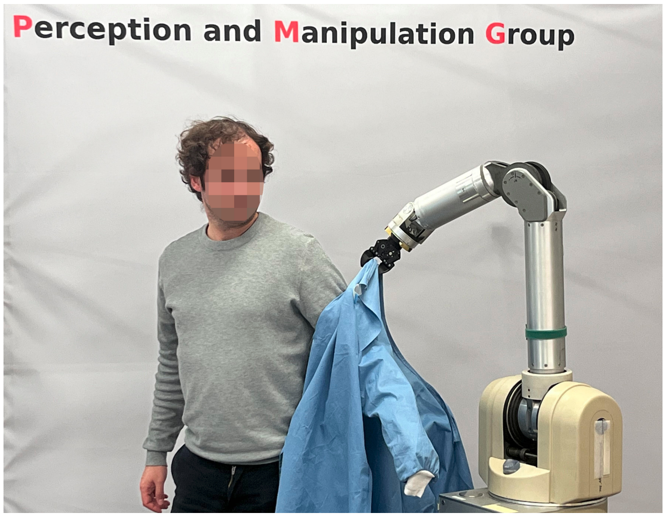

Fig. 1. A Barret's WAM robot performing an assistive robotics task of dressing a human. In these scenarios where human-robot physical interaction is present, safety measures must be prevalent and ensure that the robot will move compliantly.

However, controllers become more complex in the Cartesian space where, through a Jacobian, the desired end-effector forces can become joint torques. It is well known that the robot's singularities can result in dangerous behaviors [6] that limit the applicability of controllers in Cartesian coordinates. But since motion specifications are usually assigned in the operational space, safety in Cartesian controllers is of upmost importance.

This paper addresses some safety issues in Cartesian control, oriented at human-robot physical interaction, and proposes, from a practical perspective, a diverse set of layers of safety to ensure that the robot behavior is stable, effective, and safe. We will firstly introduce the necessary control problems in Sec. II, and then safety layers for robot learning and control in Sec. III. Some of these methods' effects are tested in Sec. IV.

II. Preliminaries

A. Robot dynamics and feed-forward control

The dynamics equation of a robot in joint space, for a certain joint configuration $\mathbf{q}$ is [7]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}_{fric} = \mathbf{u}_c - \mathbf{u}_e, \quad (1)$$

where $\mathbf{M}(\mathbf{q})$ is the inertia tensor of the robot, $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})$ is the term containing the coriolis and centripetal forces, $\mathbf{G}(\mathbf{q})$ the gravity, $\mathbf{F}_{fric}$ the robot friction, $\mathbf{u}_c$ the controller torque and $\mathbf{u}_e$ the external forces on the robot, represented by their effect in the joints. In general, $\mathbf{G}(\mathbf{q})$ can be easily estimated from the geometry and mass of the robot, while the velocity terms $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})$ are not so straight-forward to compute, but can also be obtained. Furthermore, $\mathbf{M}(\mathbf{q})$ can be easily calculated, but the accelerations $\ddot{\mathbf{q}}$ can become very noisy if they are obtained by differentiating position twice. Moreover, the friction forces are of a complex nature and can come from static and viscous friction in the joints, motors, or cables in cable-driven robots. Even magnetic friction within the motors may occur, resulting in $\mathbf{F}_{fric}$ being a black box which can be partially modelled [8], but will often be one of the main sources of dynamics modelling error.

Feed-forward controllers [7] are a common control approach [9] for human-robot interaction. In this case, an approximation of the inverse dynamics of the robot is obtained, i.e., the left side of Eq.(1). If we define the inverse dynamics as:

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) := \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}_{fric}, \qquad (2)$$

then, an approximation $\hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ will yield a modelling error of $\boldsymbol{\epsilon}_n = \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) - \hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. Now, feed-forward controllers [7] take the form:

$$\mathbf{u}_c = \hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \mathbf{u}_{\text{PID}}, \qquad (3)$$

where $\mathbf{u}_{\text{PID}}$ is a classical Proportional-Integral-Derivative controller. Now, inserting $\mathbf{u}_c$ into (1), we obtain:

$$\mathbf{u}_{\text{PID}} = \boldsymbol{\epsilon}_n + \mathbf{u}_e.$$

This equation has a high relevance as it indicates that, by using a feed-forward controller, our PID term will be responsible for compensating the dynamics modelling error $\boldsymbol{\epsilon}_n$ and the effect that external forces have on joints, $\mathbf{u}_e$, in order to track a certain trajectory. In most robotics applications, where the robot velocities and accelerations are rather small, the inertia and Coriolis effects are not the most relevant terms of the dynamics of the robot, especially in cable-driven robots such as the Barrett's WAM. Robots that have the motors on their joints do have more inertia and therefore those terms gain relevance. Nevertheless, from the terms in Eq.(2), the most relevant is usually the gravity $\mathbf{G}(\mathbf{q})$, followed by the frictions $\mathbf{F}_{fric}$, while the inertia terms are smaller for relatively small motions. Therefore, it is also common to use $\hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{G}(\mathbf{q})$ as a simplified dynamics model. Also note that the integral part of the PID controller is often omitted for robotic trajectories with variable impedance control. This is due partly to the fact that a PD controller can impose a second-order differential equation's behavior to the system (the robot) [7]. Empirically, an integral part can also be added in order to reduce steady-state error.

Up to now, we have considered control from a joint perspective, assuming that we will control the d degrees of freedom of the robot. However, in most robotic tasks, controlling the Cartesian pose (including position and orientation) is of most importance. Therefore, in the following section we develop how to extend this feed-forward control formulation to the Cartesian space.

B. Cartesian control with jacobian transpose

Jacobian transpose Cartesian control [10] is a well-known method to calculate the torques required to apply at joints to follow an end-effector (EE) pose (position and orientation) trajectory. By using the principle of virtual work on the manipulator system, it can be easily seen that the relation between the 6-dimensional Cartesian force-moment vector acting at the end-effector $\boldsymbol{f}_e$ (wrench) and the d-dimensional vector of torques at joints $\mathbf{u}$ is the transpose of the geometric Jacobian $\mathbf{J} := \mathbf{J}(\mathbf{q})$ of the manipulator:

$$\mathbf{u} = \boldsymbol{J}^T \boldsymbol{f}. \qquad (4)$$

Note that this equation allows us to work in the operational space without using any inverse kinematics. In fact, we do not need the dynamics terms in Eq. (1) expressed in the Cartesian space in order to control the robot, but instead, we will compute the error in the Cartesian space and multiply it by a Cartesian gain matrix to obtain a desired force on the end-effector:

$$\boldsymbol{f_e} = \boldsymbol{K}_P \boldsymbol{e} + \boldsymbol{K}_D \dot{\boldsymbol{e}}, \qquad (5)$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are the proportional and derivative gains matrices, and $\mathbf{e}$ and $\dot{\mathbf{e}}$ are the Cartesian position and velocity errors, including orientation if necessary. In this paper, we calculate the forces in Eq. (4) as a proportional-derivative (PD) law (see Figure 2).

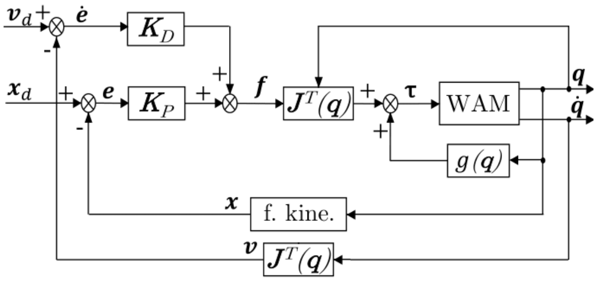

Fig. 2. Jacobian transpose control scheme.

C. Orientation error representation

As mentioned before, the geometric Jacobian is used for control, due to how easy and simple it is to compute. However, it implies that the angular velocities that are associated with it are expressed in twist coordinates. Here, we present a way to derive the orientation error associated with it. Moreover, from now on, we define the norm of a Cartesian pose vector $\|\mathbf{e}\|$ with the equivalence of 2rad=1m. Now, given that we have the current and desired orientations of the end-effector, expressed as rotation matrices R, R_d, the orientation error in twist coordinates that matches the geometric Jacobian representation is:

$$\mathbf{e}_O = \frac{1}{2}\left(R^{(1)} \times R_d^{(1)} + R^{(2)} \times R_d^{(2)} + R^{(3)} \times R_d^{(3)}\right), \quad (6)$$

where $R^{(i)}$ is the i-th column of matrix R.

D. Impedance control

Several control schemes can be used to control the manipulator executing the task. In particular, we consider Impedance Control (IC) in Cartesian space, as it is particularly suitable for the Learning from Demonstration (LfD) scenario [7]. Impedance Control, and its extension Variable Impedance Control (VIC) [11], is a popular control scheme that, combined with an inverse dynamics model as in Eq. (3), allows to obtain the following control law with a PD controller, for stiffness and damping matrices $\mathbf{K}_P$ and $\mathbf{K}_D$, reference $\mathbf{x}_d$ and measured EE position and velocity $\mathbf{x}$, $\dot{\mathbf{x}}$:

$$\mathbf{u}_{\text{PD}} = \mathbf{J}^T[\mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})]. \quad (7)$$

If $\mathbf{K}_P$ and $\mathbf{K}_D$ are diagonal, the manipulator is controlled by means of *diagonal control*. This control action, as mentioned before, can be theoretically shown to stabilize the system by means of Lyapunov functions, and imposes a second-order behavior.

III. Control features to enhance safety

As already mentioned, in this work several techniques have been implemented in order to improve safety:

A. Joint limits in kinesthetic teaching

A common method for teaching a robotic task is to set the robot in gravity compensation mode and manually guide it. This has the advantage of requiring close to no knowledge on robotics, it being an easy approach to teaching robots. However, a commonly ignored issue with this approach is the fact that, like humans, robots have their joint limits and, during kinesthetic teaching of a redundant robotic arm, a human does not usually pay much attention to the robot redundancy and might accidentally be pushing a joint towards its limit. This results in a dangerous encoding of the robot motion, which can lead to harming the hardware in execution. In order to prevent teaching trajectories too close to joint limits, a simple improvement can be done for robots with gravity compensation and variable control frameworks. Usually, kinesthetic teaching implies setting the robot in a gravity compensation mode, while deactivating any other control action. However, we have found that, in order to prevent such dangerous situations when the teaching pushes joints towards their limit, we can activate a controller that pushes the joint away from its limit. Such controller's gain $\mathbf{K}_p$ will be independent for each joint j (therefore $\mathbf{K}_p = \text{diag}(K_p^1, ..., K_p^d)$) and we will set a threshold ξ, as the distance from a joint limit in which the passive controller will push the joint away from its limits for each joint j, $\left[q_j^{\min}, q_j^{\max}\right]$. Then, we define the joint torques in kinesthetic teaching as

Case $q_j \in \left[q_j^{\max} - \xi, q_j^{\max}\right]$. We set a desired position

$$q_j^d = q_j^{\max} - \xi,$$

which results in

$$u_j = K_p^j(q_j^d - q_j).$$

Case $q_j \in \left[q_j^{\min}, q_j^{\min} + \xi\right]$. We set a desired position

$$q_j^d = q_j^{\min} + \xi,$$

which results in

$$u_j = K_p^j(q_j^d - q_j).$$

Otherwise, we deactivate this controller.

This layer of control over a theoretically non-controlled robot has an effect that prevents the robot from being too close to its joint limits while being taught and, given that anthropomorphic arms usually have one redundant degree-of-freedom (DoF), such redundancy can still allow for tracking the trajectory the human is teaching.

B. Error value saturation

It has been empirically proved that high error value signals lead to unstabilities and undesired behaviors in the robot control system. This is why a first measure consists in saturating the position and orientation error values so that they cannot be higher than certain limits, in absolute terms, for each Cartesian coordinate j. Therefore we update $\mathbf{e}$ components e_j, into a new vector $\hat{\mathbf{e}}$:

$$\hat{e}_j = \begin{cases} e_j & \text{if } |e_j| \leqslant e_j^{\max} \\ \text{sign}(e_j) \cdot e_j^{\max} & \text{otherwise} \end{cases} \quad (8)$$

This limitation on the Cartesian coordinates error has a relevant effect, as we will see in Sec. IV, of indirectly limiting the torques applied. This has a positive impact in situations like a large initial error on a trajectory, or a deviation from the desired pose due to an external effect, e.g. a human action or environmental constraint, that could otherwise result in a high-torque, high speed backlash that would not be suitable for physical human-robot interaction.

C. Time dependency on error

If motion is parametrized, e.g. with a movement primitive, the time can be slowed down with respect to error, so that error stops growing and prevents backlashing.

$$t_{k+1} = t_k + \frac{dt}{1 + \alpha \cdot \|\mathbf{e}_k\|^2} \cdot p, \quad (9)$$

where p is a flag variable that becomes 0 if the error is larger than a certain threshold, $\|\mathbf{e}\| > \epsilon_{\max}$, and 1 otherwise. Also, α is defined as:

$$\alpha = \frac{1}{2}K_\alpha\left(1 + \text{sign}(\|\mathbf{e}_k - \epsilon_{\min}\|)\right),$$

K_α being a gain factor (which we have empirically found that a suitable value for it is 100). The sign operation means that, if the error is below a certain minimal threshold $\epsilon_{\min}$, then α will be zero, and therefore $t_{k+1} = t_k + dt$. However, once the error $\mathbf{e}_k$ grows beyond $\epsilon_{\min}$, the difference between t_{k+1}

and t_k will be smaller than the usual time-step dt, therefore slowing the time used for the trajectory generator, until a point where a threshold is reached $\|\mathbf{e}_k\| = \epsilon_{\max}$, at which $p = 0$ and $t_{k+1} = t_k$. Slowing the time evolution and even stopping it has an impact on the desired pose. First, it allows the robot not to drift too far away from the reference. Imagine a robot being held by a human while it performs a motion. If the trajectory time continues to evolve, the desired pose will be increasingly further away from the current state. This means that the backlash happening once the human releases the robot can be potentially large, but also that the robot will not follow a trajectory with a shape according to what was expected, but rather try to quickly catch up to the current position. Depending on the task, one might want one or the other.

D. Elbow and wrist singularities

A manipulator is kinematically redundant when it has more DoF than the dimension of the operational space [7] - 6 in case of Cartesian control -. These extra DoF can be exploited to achieve a secondary goal while performing a task as, for instance, making the redundant joints to keep the robot as far as possible from people. To this end, a secondary controller can be constructed by projecting the secondary goal $\boldsymbol{\tau}_{null}$ into the null-space orthogonal projection of the Jacobian, with $(\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{J}^{T^\dagger})$, where $J^\dagger$ is the Moore Penrose pseudo-inverse, obtaining:

$$\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{f} + \boldsymbol{g}(\boldsymbol{q}) + (\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{J}^{T^\dagger})\boldsymbol{K}_{\text{null}}\boldsymbol{\tau}_{\text{null}}, \qquad (10)$$

where $\boldsymbol{K}_{\text{null}}$ is a diagonal matrix used to select the relevance of the secondary goal and $\boldsymbol{J}^{T^\dagger}$ is the pseudo-inverse of the transpose Jacobian matrix. A deep study on kernel projection methods can be found in [6].

In the case of serial robotic arms, it is often recommended to avoid singular positions in the robot while it is moving. Such singular positions can cause, in the Cartesian space, abrupt and fast motions that might damage the robot or its environment. It is therefore recommended to avoid such situations, and we can do so by projecting a gradient, pushing the joint pose away from the singular position using the manipulability index [12]. However, in the case of anthropomorphic arms, the singularities are known, and happen when the wrist is straight (i.e., $q_6 = 0$) or when the elbow is straight ($q_4 = 0$). In those cases, the Jacobian matrix, which is 6×7 for the case of anthropomorphic arms and usually has rank 6, becomes ill-conditioned and looses one rank. That expands the redundancy of the manipulator, which is, under normal circumstances, a rotation of the elbow point around the axis defined by the shoulder and wrist points. The added redundancy dimension is a rotation with $q_5 = -q_7$ when $q_6 = 0$ (wrist singularity) and $q_3 = -q_5$ when $q_4 = 0$ (elbow singularity). These redundancies manifest in potential vibrations of fast motions that do not affect the end-effector's pose, but can cause instability, send an error of velocity at an electronic level, or be dangerous to the surroundings.

To avoid these vibrations we propose to control the kernel of the Jacobian matrix. A torque vector $\boldsymbol{\tau}_{null}$ proportional to the gradient of a function H_w is defined. This function expresses how fast joints 5 and 7 are moving, therefore the goal is to minimize this function so the wrist joints move as minimum as possible and, hence, avoid their swap.

$$\begin{aligned} \boldsymbol{\tau}_{null} &= -\mu_4 \nabla(H_4) - \mu_6 \nabla(H_6) \\ H_4 &= (\dot{\theta}_3^2 + \dot{\theta}_5^2) \\ H_6 &= (\dot{\theta}_5^2 + \dot{\theta}_7^2)\,; \end{aligned} \qquad (11)$$

Note that regulating factors μ_4, μ_6 are defined. These variables are 1 when the robot is in the singular configuration ($q_4 = 0, q_6 = 0$, respectively) and exponentially decay to 0 when far from it:

$$\begin{aligned} \mu_4 &= \exp(-4q_4^2) \\ \mu_6 &= \exp(-4q_6^2). \end{aligned} \qquad (12)$$

This control measure mitigates the free, uncontrolled motion that might occur in the additional kernel space generated in a singularity, by minimizing the velocities in such dimension of the kernel of the Jacobian.

E. Derivative term tuning

In the previous section, we discussed how a PD controller can impose a second-order behavior on the robot's Cartesian state by modulating the proportional and derivative gains:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K_D}\dot{\boldsymbol{e}} + \boldsymbol{K_P}\boldsymbol{e} = \boldsymbol{0} \qquad (13)$$

$$t^2 + \boldsymbol{K_D} t + \boldsymbol{K_P} = 0 \qquad (14)$$

From Eq. (14), we can impose a critically damped behavior on the dynamical system by solving the ordinary differential equation and setting the eigenvalues to have no imaginary part and be negative. This comes down to imposing the discriminant $\boldsymbol{K_D}^2 - 4\boldsymbol{K_P} = 0$, which results in $\boldsymbol{K_D} = 2\sqrt{\boldsymbol{K_P}}$. This derivative gain provides a theoretical critical damping. However, the robotic platform can often have unpredictable dynamical behaviors in the transfer to reality, discretized control time, or state measurement errors. This can result in chattering in certain joints, due to the derivative term. Therefore, we recommend to use this restriction as an upper bound $\boldsymbol{K_D} \leqslant 2\sqrt{\boldsymbol{K_P}}$. This could result, theoretically, in small oscillations in the step response of the controller. However, remaining unmodelled frictions act as an additional damping on the robot that alleviate these potential oscillations.

F. Control signal saturation

Similarly to the way we proceeded in III-B, we can also set boundaries for the control signal, in order to avoid undesired jolts. Thus we can update $\mathbf{u}$ components u_i, into a new vector $\hat{\mathbf{u}}$:

$$\hat{u}_i = \begin{cases} u_i & \text{if } |u_i| \leqslant u_{i,\max} \\ \text{sign}(u_i) \cdot u_{i,\max} & \text{otherwise} \end{cases} \qquad (15)$$

This is often already considered as a safety layer at a lower level, due to the fact that the electrical currents applied on the joints have a limited amount of power that they can supply, but having control over it also improves the safety of the robot's behavior.

G. Joint limit avoidance

One last measure has been implemented in order to ensure safety. In this case, we set the corresponding torque components to 0 if the robot configuration is close to some joint limit and it is still pushing towards it. We apply this measure to as in Sec. III-A, but if, for instance, $q_j \in \left[q_j^{\max} - \xi, q_j^{\max}\right]$, we set the torque to $u_j = \min(u_j, 0)$ if $\dot{q}_j > 0$, thus not pushing away from the joint limit, but rather ensuring that the robot, if close to the joint limit and moving towards it, will not apply additional torque to such joint that would generate an acceleration towards its limit.

IV. Experimental evaluation

Dedicated experiments have been conducted to analyze the robot's behavior under the influence of some of the proposed control features. The experiments were performed in a Barrett's WAM robot (as seen in Fig. I). We used a Cartesian controller based on the described methods in Sec. II, with a compensation of the gravity term and a derivative term depending on the proportional gain as in Sec.III-E:

$$\mathbf{u}_c = \mathbf{G}(\mathbf{q}) + \mathbf{J}^T[\mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})]$$

With this base controller, we tested the terms considered throughout this paper and assessed their qualitative performance.

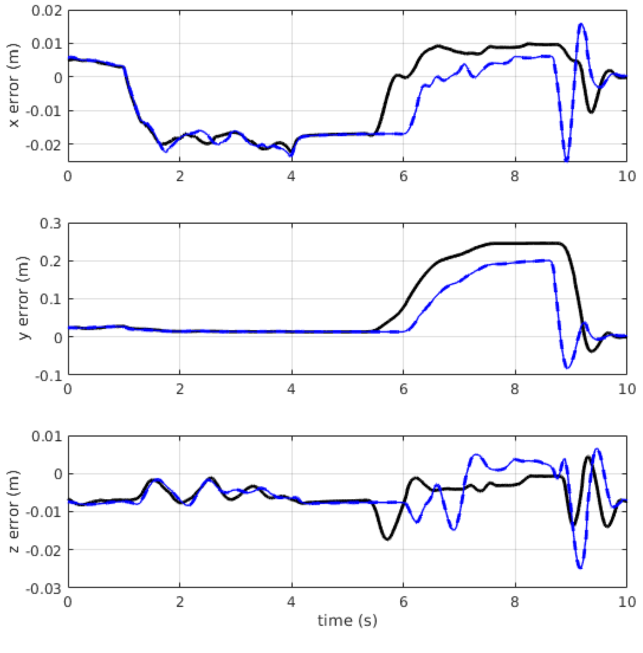

Fig. 3. Positioning error values in the Cartesian space. In black, with error saturation while, in blue, without error saturation.

A. Error value saturation

We compared the controller's performance with and without a 5cm error saturation limit. While moving the robot in a horizontal, linear trajectory, a 25cm perturbation in position was applied. Figure 3 shows the positional error in the x, y, and z components for both cases. Figure 4 displays the corresponding Cartesian forces in each direction.

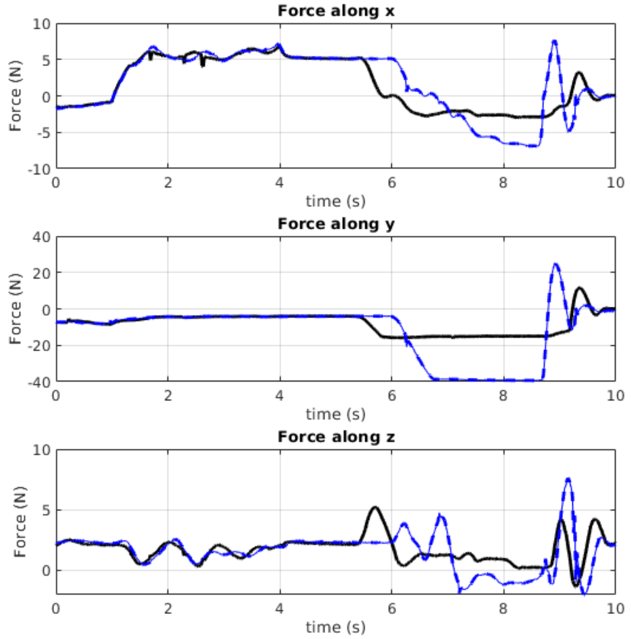

Fig. 4. Forces in the x, y, z directions in the Cartesian space. In black, with error saturation while, in blue, without error saturation.

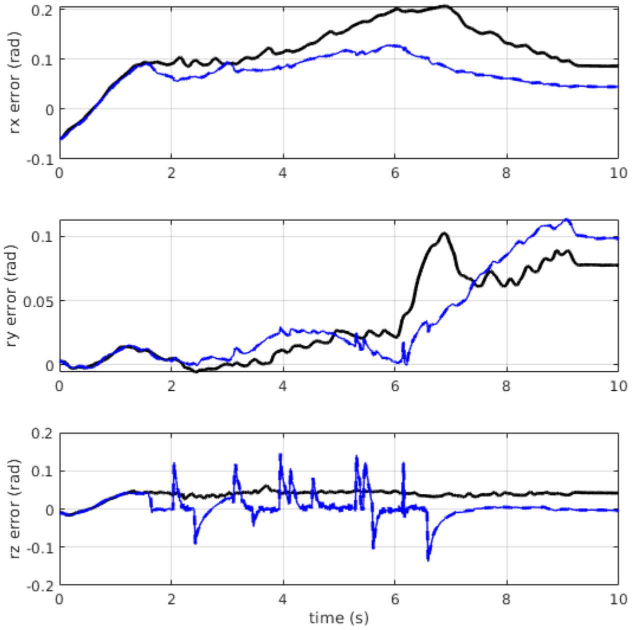

Fig. 5. Orientation error values in the Cartesian space. In black, with a reduction of the derivative term; in blue, without.

With no perturbations, the tracking error in Fig. 3 is similar between the two cases. However, the forces applied to the end-effector during perturbations are significantly larger without error saturation. For instance, in the y-component, these forces reach up to $40N$ without saturation in Fig.4, compared to less than $20N$ with the error saturation limit.

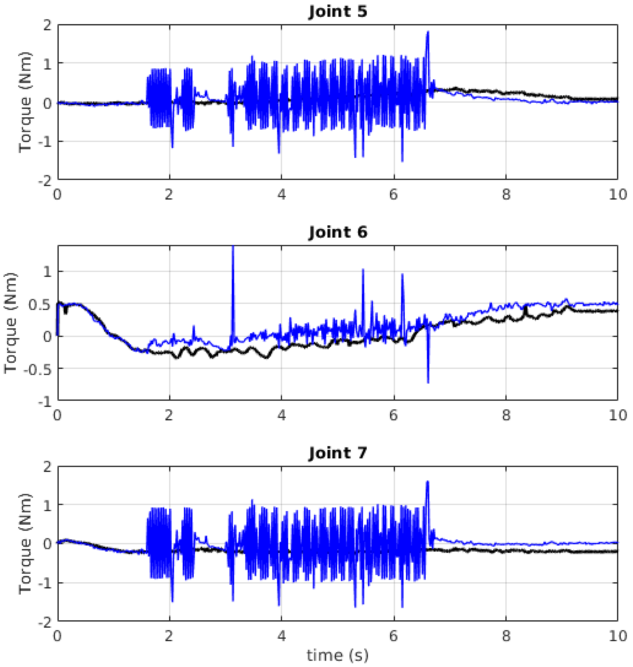

Fig. 6. Joint torque values for the wrist joints of the robotic arm. In black, with a reduction of the derivative term; in blue, without.

B. Reducing the derivative gain

In this experiment we compared the different behaviors of the robot when reducing the derivative gain below the theoretical limitation and when not. Figure 5 shows the orientation errors along a predefined L-shaped trajectory with no external disturbances. Figure 6 displays the robot's wrist joint torques for such motion. We can clearly see how, when we do not reduce the derivative gain below the limit, an undesirable high-frequency oscillation of the torque appears. Contrary to one's intuition, one would think that oscillations would occur when we reduce the derivative term but, as discussed in Sec. III-E, the oscillations from reducing such term are damped by the robot's unmodelled friction. On the contrary, a critically damped derivative term might destabilize some joints due to unmodelled phenomena.

C. Nullspace singularity control

We applied the singularity control defined in Sec. III-D, and compared its behavior against a controller which used the redundant DoF of the robot to keep the elbow in a vertical position. The performed trajectory started with the robot 15cm away from the an arbitrary initial desired position, generating a step response on the controller. Moreover, the initial joint position of the robot was with $q_6 = 0$, i.e.: in the wrist singularity position. Then, at the end of the trajectory, a human operator pushed the elbow of the arm to a horizontal position, and then released it afterwards. We also removed the error saturation safety measure, and we can observe the positioning error results of this experiment in Fig. 7 with the singularity control (in black) and with the elbow positioning in the kernel of the Jacobian (in blue). Here we see:

- When not using the singularity control and starting at a singular, or close to a singular position, a high frequency oscillation might occur in the last joints. This is due to the phenomena discussed in Sec. III-D.

- When pushing the elbow, the singularity control layer allows for a free motion of the elbow as long as we do not cross a singular position. Meanwhile, the controller with the secondary objective of keeping the elbow vertical tries to prevent that motion, and therefore increases the end-effector error. Moreover, due to friction and the kernel force colliding with the main end-effector task, the robot has a larger error in steady state once released.

V. Conclusion

In this work, we identified common challenges encountered when implementing Cartesian control on redundant robotic arms, specifically using a 7-DoF Barrett WAM manipulator as a case study. We introduced the Cartesian control framework in a tutorial manner, and provided tools to enhance safety. First, artificial forces in a kinesthetic teaching for preventing situations that are *too close* to a danger zone. Also, a way of modulating the derivative of the control timer so as to slow trajectories in execution and prevent backlashes when a contact occurs. Then, we tested how saturating the error values in addition to joint torque values can help a controller to prevent large forces, especially in cases where the initial desired position is not the current initial position. We have also shown how a reduction of the derivative gain can reduce oscillations, and a nullspace control of the redundancy prevents the free motion in the nullspace to be too fast and, while the theoretical derivative term of a PD controller can be obtained analytically, it is shown in this paper that a smaller value might actually be safer for real robot executions.

Our experimental results show how the robot responds with and without these safety features, highlighting that some are universally necessary while others depend on the specific task. Future work will focus on developing a hierarchical safety control taxonomy to guide the selection of appropriate safety capabilities based on task requirements.

Acknowledgment

This work was partially funded by CSIC Project 202350E080 ClothIRI: Robotic Cloth Manipulation at IRI.

References

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

[2] M. Iskandar, G. Quere, A. Hagengruber, A. Dietrich, and J. Vogel, "Employing whole-body control in assistive robotics," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5643–5650, 2019.

[3] O. S. Ajani and S. F. Assal, "Hybrid force tracking impedance control-based autonomous robotic system for tooth brushing assistance of disabled people," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 4, pp. 649–660, 2020.

[4] M. Bogdanovic, M. Khadiv, and L. Righetti, "Learning variable impedance control for contact sensitive tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6129–6136, 2020.

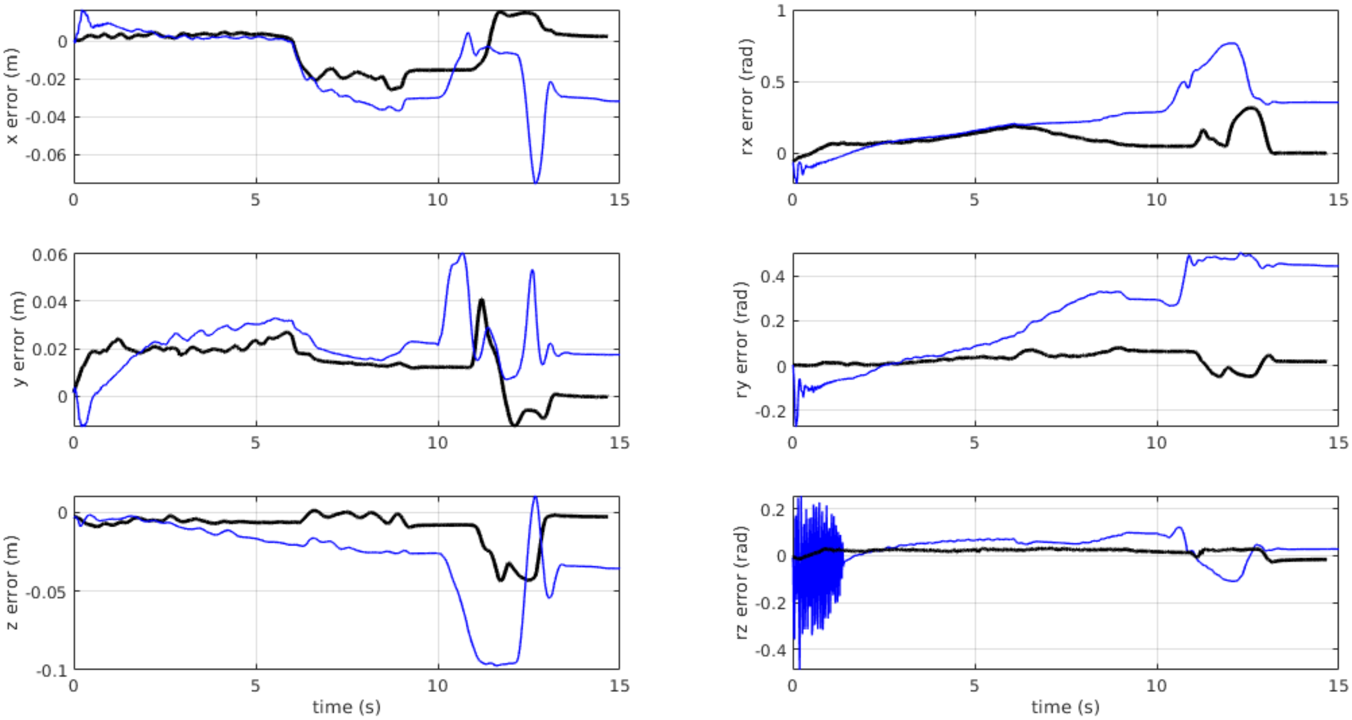

Fig. 7. Position and orientation error values in the Cartesian space. In black, with singularity control; in blue, with the elbow positioning in the kernel of the Jacobian.

[5] S. A. Schwarz and U. Thomas, "Variable impedance control for safety and usability in telemanipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6177–6182, 2022.

[6] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 944–955, 2015.

[7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st ed., 2008.

[8] A. Colomé, A. Planells, and C. Torras, "A friction-model-based framework for reinforcement learning of robotic tasks in non-rigid environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5649–5654, 2015.

[9] A. Colomé, D. Pardo, G. Alenyà, and C. Torras, "External force estimation during compliant robot manipulation," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3535–3540, 2013.

[10] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1989.

[11] F. J. Abu-Dakka and M. Saveriano, "Variable impedance control and learning—a review," *Frontiers in Robotics and AI*, vol. 7, 2020.

[12] T. Yoshikawa, "Dynamic manipulability of robot manipulators," *Transactions of the Society of Instrument and Control Engineers*, vol. 21, no. 9, pp. 970–975, 1985.