2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Online velocity constraint adaptation for safe and efficient human-robot workspace sharing

Lucas Joseph[1], Joshua K. Pickard[1], Vincent Padois[1] and David Daney[1]

Abstract— **Despite the many advances in collaborative robotics, collaborative robot control laws remain similar to the ones used in more standard industrial robots, significantly reducing the capabilities of the robot when in proximity to a human. Improving the efficiency of collaborative robots requires revising the control approaches and modulating online and in real-time the low-level control of the robot to strictly ensure the safety of the human while guaranteeing efficient task realization. In this work, an openly simple and fast optimization based joint velocity controller is proposed which modulates the joint velocity constraints based on the robot's braking capabilities and the separation distance. The proposed controller is validated on the 7 degrees-of-freedom Franka Emika Panda collaborative robot.**

I. Introduction

Improved processes, increased flexibility, reduced production footprint, and the preservation and valorization of workers' health and expertise are among the most cited promises of collaborative robots for the industry [1]. These promises raise high expectations for collaborative robots, and despite the emergence of technological breakthroughs such as joint torque sensing [2], collaborative robot control laws remain very similar to the ones used in more standard industrial robots, just accounting for standards dedicated to collaborative robots without deeply revising the control approaches themselves.

The ISO 10218-1:2011 standard on safety requirements for industrial robots [3] and ISO/TS 15066:2016 standard for collaborative robots [4] have led to the emergence of two approaches for safety in workspace sharing contexts. The first one is known as Speed Separation Monitoring and the second one as Power and Force Limiting. While the later accounts for the transient and steady-state forces that can be safely applied to the human body during physical contact, the former uses an estimation of the robot stopping time to provide safe separation distance computation rules as a function of several other factors [5], [6]. In both cases, applying these safety rules implicitly requires a knowledge of the robot breaking capabilities. These capabilities are not always provided by the robot manufacturer and, in practice this leads to collaborative features restricted to discrete modulation of the robot velocity as a function of the presence of humans in predefined zones, triggered safety stops and automatic restart without the need for human intervention in case of a controlled stop.

As a consequence, while most of the 2D laser ranger sensors used to define discrete zones for safety also provide "continuous" information regarding the distance of the closest obstacle, safety in existing industrial environments is envisioned either as an *a priori* feature, *i.e.*, the robot trajectory is adapted as a function of the considered safety zone, or *a posteriori*, *i.e.*, controlled emergency stops are triggered in case some safety limits on the separation distance are exceeded. Safety zones can, for example, be implemented as in [7], [8] and used to stop the robot or to reduce the robot velocity [9].

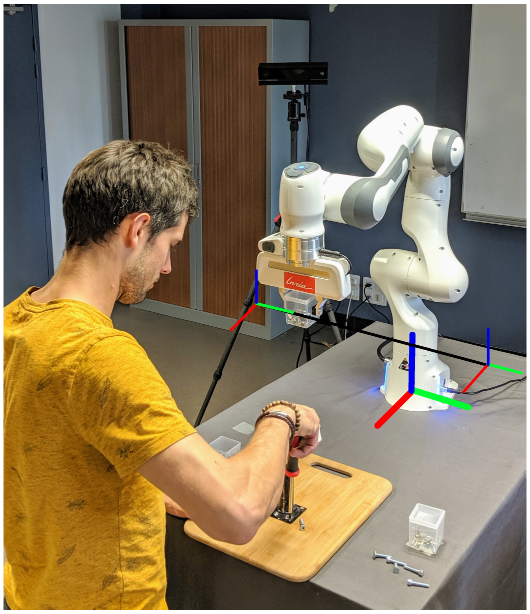


Fig. 1. Human-robot workspace sharing when performing collaborative tasks showing the Franka Emika Panda collaborative robot, the Microsoft Kinect sensor, and the desired trajectory generated offline.

In order to achieve more efficient behaviours, safety must be considered at the control level, by modulating **online** the **real-time low-level control** of the robot. Such online control law adaptations are explored in the literature. For example, safety can be ensured by computing a set of control inputs that respect a threshold on some safety markers and the work in [10] proposes to compute a set of torque commands that respect a limit on the impact force. In [11], a single axis system adapts its reference trajectory to prevent the accumulation of energy during contact. Some of the authors of this paper have also proposed control approaches [12], [13] where explicit constraints on the robot energy are set at the control level. All of the proposed solutions prevent, to some extent, the robot from being hazardous during an undesired interaction with a human. However, the methods used to lower the robot performances do not guarantee an optimal achievement of the

[1] Auctus, Inria - IMS (Univ. Bordeaux / Bordeaux INP / CNRS UMR 5218), 33405 Talence, France `firstname.name@inria.fr`

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

desired tasks. Also, while promising, these approaches cannot be applied directly to velocity controlled robots and are thus still unfit for the industrial world.

In this work, the proposed control approach is openly simple in its principle, and its implementation is straightforward. It relies on a modulation of constraints on the joint velocities to strictly ensure that the robot is capable of completely braking before collision with a human. This controller, which draws its inspiration from the work in [14], is formulated as an optimization based controller which returns safe control actions **without the need for explicitly pre-defining safety distances**. This approach makes use of specifications provided by the robot manufacturer that describes the capabilities of the robot throughout its workspace. With the assumption that these specifications describe the worst-case capabilities of the robot, they can be used to determine the robot's braking capabilities. The proposed concept of braking capabilities of a robot describes its ability to reduce all of its joint velocities to zero. For a given stopping time, which can be derived from the separation distance and velocity of the human, joint velocity limits can be derived which ensure that the current joint velocities can be driven to zero before the human reaches the robot. Thus, the robot's performance is maximized without jeopardizing the human's safety.

The paper is organized as follows. Section II presents an approach to select the joint velocity limits based on the robot's braking capabilities. Section III defines the optimization based controller used to constrain the joint velocity commands sent to the robot actuators. To demonstrate the performance of the approach, the proposed controller is implemented on the 7 degrees-of-freedom (dof) Franka Emika Panda collaborative robot. Section IV consists of experimental results exposing the interesting properties of the proposed algorithm. Finally, Section V presents a discussion on several aspects of the controller and advocates for a better access to and use of robot joint level capabilities.

II. Braking capabilities

A safe human-robot interaction, such as the task shown in Fig. 1, must ensure that the robot's kinetic energy dissipated in case of a physical contact cannot cause harm to the human. Since the kinetic energy, E_k, of a n-dof robot is related to its joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ through

$$E_k = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the mass matrix evaluated in configuration $\boldsymbol{q} \in \mathbb{R}^n$, modulating joint velocities has a similar effect to modulating kinetic energy.

Given the separation distance between the human and the robot as d_s and the maximum linear velocity of the human as v_h, let the minimum time to collision be $t_{\text{col}} = d_s/v_h$. The robot is considered to be safely operating if it is able to dissipate its kinetic energy below a safe limit, taken simply as 0 Joules in this work, within a time of t_{col}.

The joint accelerations and jerks are defined by $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ and $\dddot{\boldsymbol{q}} \in \mathbb{R}^n$ respectively. In the worst-case, as depicted in Fig. 2, the robot is currently moving with a maximum velocity of $-\dot{\boldsymbol{q}}_{\max}$ and a maximum acceleration of $-\ddot{\boldsymbol{q}}_{\max}$. In order to reduce the joint velocities to zero, the robot can accelerate at a maximum jerk of $\dddot{\boldsymbol{q}}_{\max}$ until the maximum acceleration $\ddot{\boldsymbol{q}}_{\max}$ is reached, maintain a constant acceleration of $\ddot{\boldsymbol{q}}_{\max}$, then decelerate at maximum jerk until $\dot{\boldsymbol{q}} = \ddot{\boldsymbol{q}} = \boldsymbol{0}$. The gray area on Fig. 2 corresponds to the evolution of the robot velocity, $\dot{\boldsymbol{q}}$, over time t_{stop}, which is the time required for the robot to stop. Since the velocity contributions of the first two sloped regions cancel out, the velocity can be expressed as

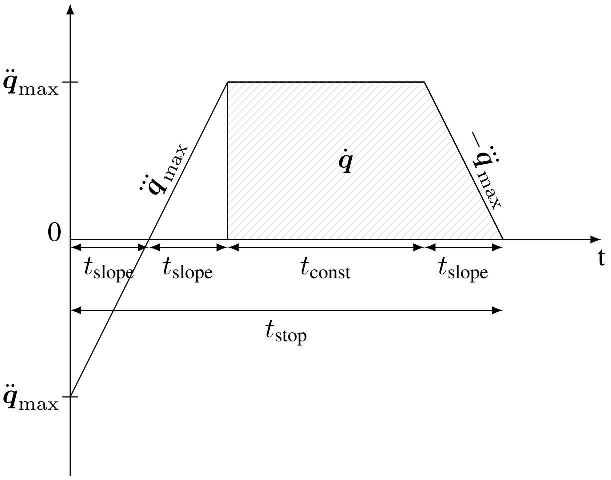


Fig. 2. Worst case ($\dot{\boldsymbol{q}} = -\dot{\boldsymbol{q}}_{\max}$, $\ddot{\boldsymbol{q}} = -\ddot{\boldsymbol{q}}_{\max}$) maximum deceleration profile until reaching a complete stop.

$$\dot{\boldsymbol{q}} = \frac{1}{2}t_{\text{slope}}\ddot{\boldsymbol{q}}_{\max} + t_{\text{const}}\ddot{\boldsymbol{q}}_{\max} \tag{2}$$

with $t_{\text{slope}} = \ddot{\boldsymbol{q}}_{\max}/\dddot{\boldsymbol{q}}_{\max}$. The overall stopping time is then $t_{\text{stop}} = 3t_{\text{slope}} + t_{\text{const}}$ and from Eq. (2), considering absolute joint velocities, the time for the robot to stop is the worst-case stopping time for all joints, given as

$$t_{\text{stop}} = \max_{i=1\ldots n}\left(\frac{|\dot{q}_i|}{\ddot{q}_{i,\max}} + 2.5\frac{\ddot{q}_{i,\max}}{\dddot{q}_{i,\max}}\right), \tag{3}$$

where i corresponds to the index of the joint.

The controller time period and the distance sensor acquisition time are respectively defined by Δt and t_{acq}. The actual time which can be allocated to stop is then given by

$$t'_{\text{stop}} = \max(0, t_{\text{col}} - \Delta t - t_{\text{acq}}). \tag{4}$$

Using Eq. (3) with $t_{\text{stop}} = t'_{\text{stop}}$, the maximum absolute joint velocities which can be reduced to zero, denoted by $|\dot{\boldsymbol{q}}_{\text{brake}}| \in \mathbb{R}^n$, is given by

$$|\dot{q}_{i,\text{brake}}| = \max(0, \min(\dot{q}_{i,\max}, (t'_{\text{stop}} - 2.5\frac{\ddot{q}_{i,\max}}{\dddot{q}_{i,\max}})\ddot{q}_{i,\max})) \tag{5}$$

for joints $i = 1, \ldots, n$.

The safe joint velocity limits which ensure braking are given by the vector of intervals

$$[\dot{\boldsymbol{q}}_{\text{safe}}] = [-|\dot{\boldsymbol{q}}_{\text{brake}}|, |\dot{\boldsymbol{q}}_{\text{brake}}|]. \tag{6}$$

These limits ensure that the robot's joint velocities can be reduced to zero within the time t'_{stop} and never exceed the maximum joint velocity limits. Considering the sensor acquisition times, braking is implicitly enforced on joint i when $t_{\text{col}} \leq t_{\text{stop}} + \Delta t + t_{\text{acq}}$.

III. Optimization-Based Controller

Although safety can be enforced *a priori* by redefining the trajectory online, it should also be enforced at the control level. First, in some applications a desired trajectory is not always determined beforehand (*e.g.*, with sensor based control or tele-manipulation). Second, one cannot ensure that the desired trajectory can be correctly tracked which may lead to the actual joint velocity required to properly reduce the tracking error exceeding the safe joint velocity limits.

The end-effector pose and the operational space twist are respectively denoted by $\boldsymbol{x} \in SO(3) \times \mathbb{R}^3$ and $\boldsymbol{v} \in \mathbb{R}^6$. Regardless of how the trajectory is generated, let's consider a desired pose $\boldsymbol{x}_{\text{traj}}$ along the trajectory. At the joint velocity level, a proportional controller with a feed-forward term can be defined as

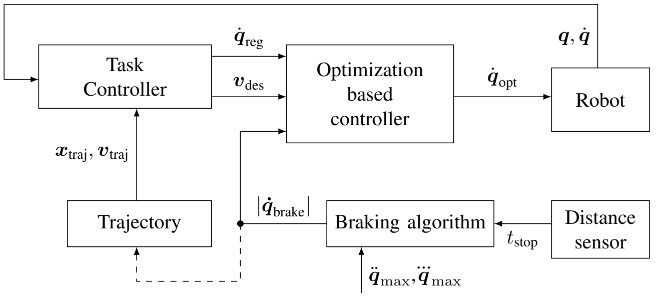
$$\boldsymbol{v}_{\text{des}} = \text{diag}(\boldsymbol{k}_p)(\boldsymbol{x}_{\text{traj}} \ominus \boldsymbol{x}) + \boldsymbol{v}_{\text{traj}} \tag{7}$$

where $\boldsymbol{v}_{\text{des}}$ is the desired velocity to correct the tracking error, $\text{diag}(\boldsymbol{k}_p)$ is a diagonal matrix of the proportional gains, $\boldsymbol{x}$ is the current pose, $\ominus$ denotes the difference between two poses in Cartesian space, and $\boldsymbol{v}_{\text{traj}}$ is the velocity required by the trajectory. The robot operational velocity $\boldsymbol{v}$ is linked with the robot joint velocity through the Jacobian, $J(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$, such that $\boldsymbol{v} = J(\boldsymbol{q})\dot{\boldsymbol{q}}$. From a control point of view the robot task can be defined as finding a set of joint velocities, $\dot{\boldsymbol{q}}_{\text{des}}$ such that

$$\boldsymbol{v}_{\text{des}} = J(\boldsymbol{q})\dot{\boldsymbol{q}}_{\text{des}}. \tag{8}$$

To ensure safety, one must ensure that the joint velocity commands sent to the robot do not exceed the joint braking velocities computed in Eq. (5). These velocities must therefore be expressed as constraints in the control problem. Two different methods exist for this. The first one consists in projecting solutions in the null-space of the robot main task in order to push the robot away from its joint velocity limits (see [15]). The second one consists in expressing the control problem as a constrained optimization problem. The optimization cost function minimizes an error between a desired operational velocity and the robot's current operational velocity. The maximum joint velocities that the robot can get are then expressed as a constraint inside the optimization problem. This method allows to track the desired position optimally while respecting the set of given constraints. The ability of the constrained optimization approach to minimize tracking error while also strictly respecting safe joint velocity limits makes it ideal for this work.

To account for the redundancy of a robot and the potentially infinite number of solutions to Eq. (8), a regularization task is considered. This auxiliary task allows determining joint velocities which can, for example, help in minimizing the error between the current joint angles $\boldsymbol{q}$ and the regularization joint angles $\boldsymbol{q}_{\text{reg}}$ as follows



$\dot{\boldsymbol{q}}_{\text{reg}}$
$\boldsymbol{q}, \dot{\boldsymbol{q}}$
Task Controller
$\boldsymbol{v}_{\text{des}}$
Optimization based controller
$\dot{\boldsymbol{q}}_{\text{opt}}$
Robot
$\boldsymbol{x}_{\text{traj}}, \boldsymbol{v}_{\text{traj}}$
$|\dot{\boldsymbol{q}}_{\text{brake}}|$
Trajectory
Braking algorithm
Distance sensor
t_{stop}
$\ddot{\boldsymbol{q}}_{\max}, \dddot{\boldsymbol{q}}_{\max}$

Fig. 3. Optimization based controller.

$$\dot{\boldsymbol{q}}_{\text{reg}} = \text{diag}(\boldsymbol{k}_{p_{\text{reg}}})\left(\boldsymbol{q}_{\text{reg}} - \boldsymbol{q}\right) \tag{9}$$

where $\dot{\boldsymbol{q}}_{\text{reg}}$ is the regularization joint velocities required to correct joint regularization errors and $\boldsymbol{k}_{p_{\text{reg}}}$ is the regularization proportional gains. In addition, $\omega_{\text{reg}} \ll 1$ is a regularization weight that provides an incentive to adjust the joint angles towards the regularization joint angles $\boldsymbol{q}_{\text{reg}}$ while minimally affecting the achievement of the main task.

The optimization problem can be written as

$$\begin{array}{lll} \dot{\boldsymbol{q}}_{\text{opt}} = & \underset{\dot{\boldsymbol{q}}}{\arg\min} & \|\boldsymbol{v}_{\text{des}} - J\dot{\boldsymbol{q}}\|^2 + \omega_{\text{reg}}\|\dot{\boldsymbol{q}}_{\text{reg}} - \dot{\boldsymbol{q}}\|^2 \\ & \text{s.t.} & \boldsymbol{q} \in [\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}] \\ & & \dot{\boldsymbol{q}} \in [\dot{\boldsymbol{q}}_{\text{safe}}]. \end{array} \tag{10}$$

The optimization output, $\dot{\boldsymbol{q}}_{\text{opt}}$, yields the optimal solution for the joint velocities and is sure to comply with the safe joint velocity limits. Whenever necessary, the velocity constraint induces an implicit deceleration of the robot using the maximum joint acceleration and jerk capabilities to ensure the safety of the human. One advantage of this formulation is that implicit braking allows to maintain a trajectory tracking task at reduced velocity while explicit braking in joint space would require giving up the trajectory tracking completely.

The proposed optimization based controller is outlined in Fig. 3. The task controller block receives joint values from the robot and the desired pose and operational velocities from the trajectory and evaluates Eq. (7) and Eq. (9). The distance sensor block determines the separation distance between the human and the robot and outputs the required stopping time of the robot. The stopping time, along with the robot's acceleration and jerk limits are fed into the braking algorithm block which evaluates Eq. (5). The optimization based controller block solves the optimization problem in Eq. (10) and sends the control action $\dot{\boldsymbol{q}}_{\text{opt}}$ to the robot. The trajectory may be precomputed offline and optionally may receive $|\dot{\boldsymbol{q}}_{\text{brake}}|$ as an input allowing for an online adaptation for modulating the velocity of the trajectory according to the current capabilities of the robot. The trajectory may also be defined online, *e.g.*, through tele-manipulation, allowing the next desired pose to be updated live (see the attached video).

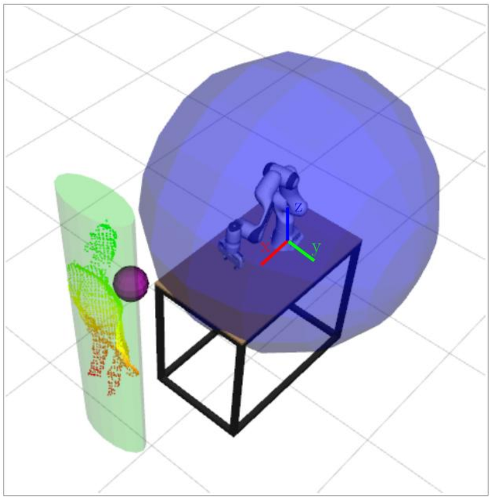


Fig. 4. The human point-cloud and associated circumscribed ellipse and the robot workspace used to calculate separation distance. The purple sphere near the human is the nearest point to the robot.

IV. Experiment

The experimental setup used in this paper is shown in Fig. 1. It consists of the 7-dof Franka Panda robot, a human subject, and a Microsoft Kinect 2 sensor[1] to capture the human's movement. Franka Emika's ROS integration packages [16] and the dynamic model provided in [17] are used together with the KDL library [18] to implement the proposed controller on the Franka Panda robot. The kinect human tracking software from [19] is used to collect a human point-cloud and determine its distance relative to the robot. The optimization algorithm in Eq. (10) is solved using the qpOASES quadratic programming solver [20]. An Intel NUC model NUC6i3SYK running a realtime Linux kernel executes the controller with a controller time period of $\Delta t = 0.001$ s. The proposed controller is executing each update in approximately $65\ \mu$s.

The Kinect sensor is calibrated with respect to the robot base. As depicted in Fig. 4, it captures a human point-cloud of filtered points on the visible surface of the human. The human point-cloud is projected to the XY plane and fit with a circumscribed ellipse. The robot's workspace is also projected to the XY plane, and a circumscribed circle is fit to the projected workspace. The separation distance between the human and the robot, d_s, is then determined by finding the minimum distance between the human ellipse and the robot circle. At all times it is assumed that the human can move at a maximum velocity $v_h = 1.6$ m/s, as specified in [4]. The Kinect sensor acquisition time is $t_{\text{acq}} = 0.03$ s, allowing the stopping time to be determined from Eq. (4).

The symmetric joint limits published by the robot manufacturer are given in Table I. These published values are assumed to be representative of the common limits throughout the robot's workspace and are therefore representative of the robot's braking capabilities. From Eq. (3), the worst-case stopping time considering maximum joint velocities is 0.295 s for joint 2. The controller gains used by the controller are given in Eq. (11). In addition, the regularization joint angles are specified as $\boldsymbol{q}_{\text{reg}} = (\boldsymbol{q}_{\max} - \boldsymbol{q}_{\min})/2$ with a regularization weight of $\omega_{\text{reg}} = 1.0e^{-7}$. The controller gains are set as

TABLE I

Common joint limits provided for the Franka Panda.

joint	$\boldsymbol{q}_{\min}$ (rad)	$\boldsymbol{q}_{\max}$ (rad)	$\dot{\boldsymbol{q}}_{\max}$ (rad/s)	$\ddot{\boldsymbol{q}}_{\max}$ (rad/s^2)	$\dddot{\boldsymbol{q}}_{\max}$ (rad/s^3)
1	-2.8973	2.8973	2.175	15	7500
2	-1.7628	1.7628	2.175	7.5	3750
3	-2.8973	2.8973	2.175	10	5000
4	-3.0718	-0.0698	2.175	12.5	6250
5	-2.8973	2.8973	2.610	15	7500
6	-0.0175	3.7525	2.610	20	10000
7	-2.8973	2.8973	2.610	20	10000

$$\begin{aligned} \boldsymbol{k}_p &= (20, 20, 20, 40, 40, 40)\ s^{-1}, \\ \boldsymbol{k}_{p_{reg}} &= (5, 5, 5, 5, 5, 5)\ s^{-1}. \end{aligned} \tag{11}$$

The desired trajectory is given as a simple linear motion generated offline and is depicted in Fig. 1. It consists of two linear segments parallel to the Y axis, each with trapezoidal velocity profiles which start and end with zero velocity, and accelerate at 0.8 m/s^2 with a maximum velocity of 0.6 m/s. The trajectory is modulated online by using $|\dot{\boldsymbol{q}}_{\text{brake}}|$ to scale the time to the next desired pose $\boldsymbol{x}_{\text{traj}}$ along the trajectory.

To demonstrate the braking performance of the proposed approach, a human subject is asked to move towards the operating robot. Data from the robot is provided in Fig. 5 which depicts the effect of separation distance / time to collision on the safe joint velocity constraints. As the human approaches the robot the minimum time to collision, t_{col}, decreases, which also decreases the minimum stopping time for the robot. This results in the safe joint velocity limits $[\dot{\boldsymbol{q}}_{\text{safe}}] = [-|\dot{\boldsymbol{q}}_{\text{brake}}|, |\dot{\boldsymbol{q}}_{\text{brake}}|]$ shrinking towards $\mathbf{0}$ rads/s. Correspondingly, the maximum time required for the robot to stop, given by Eq. (4), also decreases.

Next, to further explore the performances of the proposed approach, a human subject is asked to work in proximity to the operating robot executing the same linear trajectory as previously described. The relationships between separation distance, desired operational velocity, trajectory tracking error and the constrained joint velocities of joints 1 and 3 are provided in Fig. 6. The plots are divided into three sections: ①, ②, and ③. In ①, although the human is less than 0.4 m from the robot, the joint velocities required to perform the task are far from the safe joint velocity constraints. Consequently, the robot is able to continue operating normally with minimal tracking error. In such situations, the trajectory modulation is not applied, and the desired linear operational velocity comes from the trapezoidal velocity profile defined in the original trajectory and the correction of the tracking error. In ②, the human moves even closer (less than 20 cm). This causes the controller to determine an optimal solution for the joint velocities that respects the safe joint velocity constraints while minimizing tracking and joint regularization error. Consequently, the robot overall motion is slower, ensuring its motion

[1] in a more industrial setting, a 2D laser ranger sensor could be used and provide more robust information.

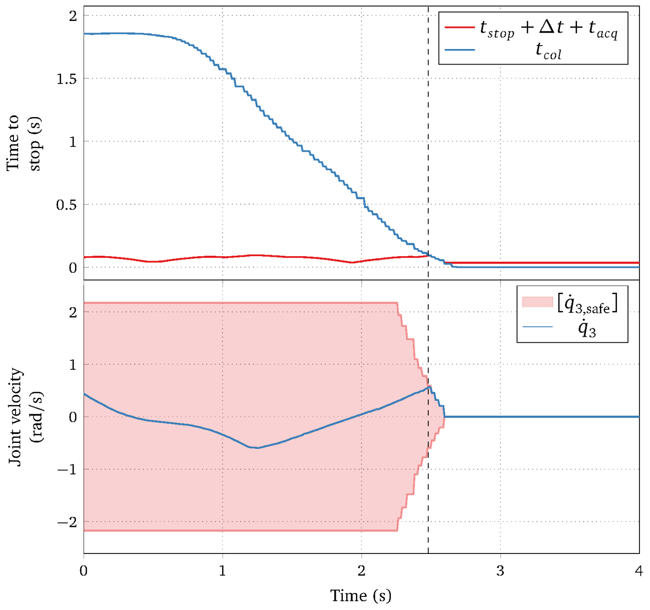


Fig. 5. The top plot provides in blue the evolution of the time to collision (time until the human enters the robot workspace) as well as the evolution of the robot stopping time (red). The bottom plot provides the evolution of the velocity of joint 3, the most critical joint in this particular experiment. The red area represents the safe velocity computed from Eq. (5). The vertical line displays the time at which the robot needs to decelerate in order to be able to fully stop before the human reaches its workspace (*i.e.*, $t_{\text{col}} \leq t_{\text{stop}} + \Delta t + t_{\text{acq}}$).

completely stops in case the human comes even closer. It can be observed as a side effect that the tracking error increases. This is due to some of the joint velocity constraints becoming active (*e.g.*, joint 3), making it infeasible for the robot to bring the tracking errors to zero. This shows that the proposed approach to modulating the trajectory can be improved. Finally, in ③ the human subject moves away from the robot. The joint velocity constraints expand, and the robot is then able to return to normal operation with minimal tracking error.

The experimental results presented in this section are displayed in a movie attached with this paper. This movie also presents a simple application where a human is working near a robot which is tele-manipulated to perform a simple pick-and-place task. This additional experiment, that is not presented in this paper, emphasizes the fact that the proposed controller is trajectory independent.

The straightforward approach presented in this paper to ensure safety can easily be applied on any velocity actuated robot as long as one can get the human separation distance and has access to the robot joint limits. The results displayed in this section show that the proposed controller only stops the robot when it is absolutely required, *i.e.* just before a collision between a human and the workspace occurs. Thus, the performance of the robot is maximized without jeopardizing the safety of the human.

V. Discussion

Throughout this paper, the robot maximum joint acceleration and jerk capabilities are considered as constants, ac-

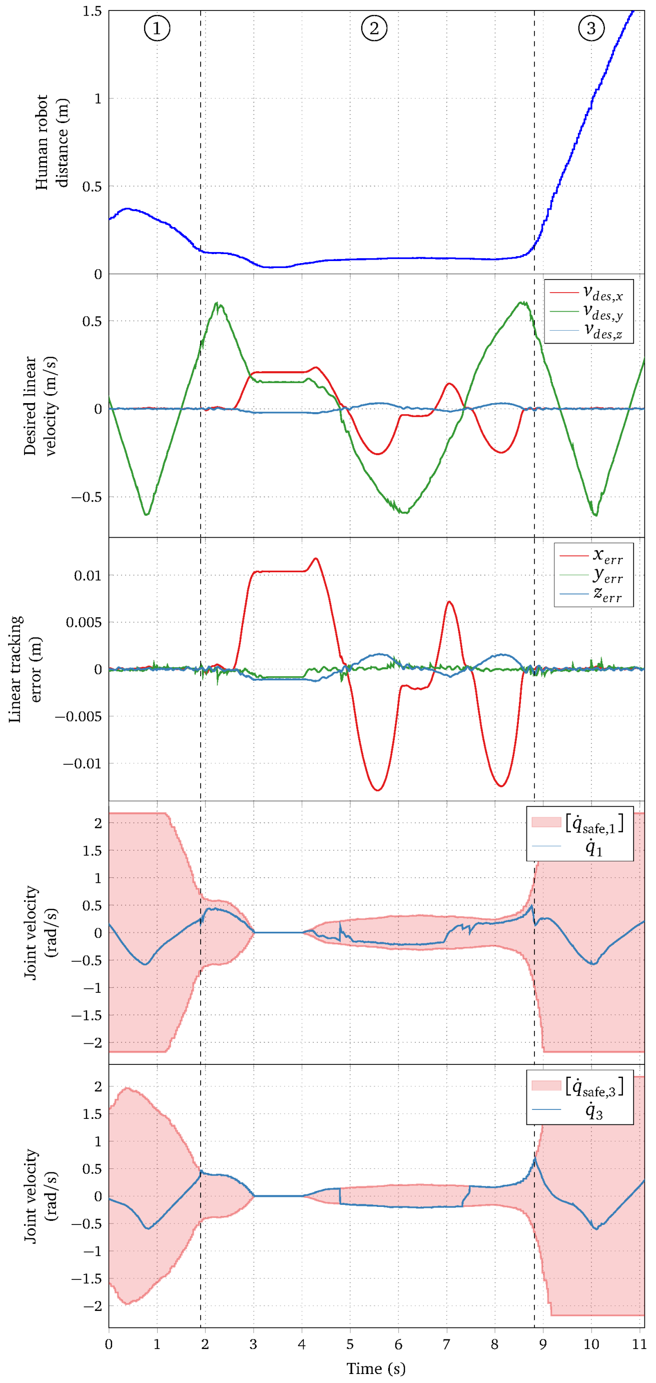


Fig. 6. Human-robot separation distance, desired operational velocity ($\boldsymbol{v}_{\text{des}}$), trajectory tracking error and joint velocity constraints of joints 1 and 3 versus time. The red area represents the safe velocities for the joints. The plots are divided in three parts: ① Where the human is far enough from the robot so that it can operate normally. ② Where the human moves closer to the robot. ③ Where the human moves away from the robot.

cording to the specifications of the robot manufacturer. These constant values are likely obtained by sampling the robot workspace for different states and taking the worst case. However, whether the robot carries a load or not greatly influences these limits. Also, the robot capabilities depend on the robot state and dynamics [21], [22] and the real technological restrictions at the joint level are on the joint position and velocity as well as on the joint torque $\boldsymbol{\tau}$ and its time derivative (these values are also provided by Franka Emika and some other manufacturers). Thus, the constant capabilities specified by the manufacturer in terms of joint jerk and accelerations cannot accurately reflect the current capabilities of the robot. Using these published capabilities to predict the future state of the robot leads in the best case, assuming these acceleration capabilities are actually underestimated, to a sub-optimal use of the actual robot capabilities.

This sub-optimality is accepted in standard industrial robotics, where robots are generally oversized, as it clearly simplifies trajectory generation and control. In collaborative robotics, actuation is designed for a given type of tasks with the sub-goal of minimizing the total weight and power of the robot for safety reasons. As a consequence, one can no longer simply use an approximated subset of the actual robot capabilities as this subset is very likely to be small and lead to really poor performances of the system. More particularly, in workspace sharing contexts this sub-optimal use of the robot capabilities implies conservative stopping behaviour of the robot which can be critical from a strictly industrial point of view. Moreover, as can be witnessed in the tracking error in Fig. 6, the tracking error increases when the joint velocity constraints become active. If path following is of importance for the applicative context, then one may want to minimize the path tracking error. This requires to adapt the desired trajectory as a function of the actual robot capabilities.

Therefore, the performance and reliability of the proposed controller and the corresponding trajectory modulation could be further improved by including state and dynamic dependant models of the robot's maximal joint acceleration and jerk. This would provide more accurate braking capabilities that do not rely on assumptions on the manufacturer's specifications and more optimal modulation of the trajectory to ensure minimal tracking error.

Actually, given the set of robot joint states $\{\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}\}$, the set of corresponding joint accelerations can be computed as a function of the torque derivative limits. Let the current joint accelerations be computed from the configuration-space dynamic model

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})^{-1} \left(\boldsymbol{\tau} - \boldsymbol{v}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right) \tag{12}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the joint space inertia matrix and $\boldsymbol{v}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\boldsymbol{g}(\boldsymbol{q})$ are respectively the Coriolis and centrifugal and gravity induced joint torques. Considering the joint torque derivative limit $\dot{\boldsymbol{\tau}}_{\max}$, the acceleration capabilities are given by the convex polytope

$$\mathcal{P} = \text{conv}\left(\{\ddot{\boldsymbol{q}} \mid \ddot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})^{-1}\text{vert}\left([\boldsymbol{\tau}_{\text{eff}}]\right)\}\right) \tag{13}$$

where

$$[\boldsymbol{\tau}_{\text{eff}}] = [\boldsymbol{\tau}_{\text{lim}}] \cap (\boldsymbol{\tau} + [-\dot{\boldsymbol{\tau}}_{\max}, \dot{\boldsymbol{\tau}}_{\max}]\Delta t - \boldsymbol{v}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}))$$

is a convex set with $[\boldsymbol{\tau}_{\text{lim}}]$ the torques limits provided by the manufacturer, conv is the convex hull, and vert extracts the vertices of the convex set.

Assuming that the torque derivative limits provided by the manufacturer are accurate, one can thus have a very good knowledge of the robot acceleration capabilities in a given state. One of the future challenges is thus to account for these realistic, state dependant, joint limits when predicting the braking capabilities of the robot. While the concept of *safety maps* introduced by N. Mansfeld *et al* [23] is promising, it clearly cannot be used as such for real-time control. The main difficulty is that this approach relies on sampling the robot capabilities throughout the workspace. This is challenging given the dimension of the state space which is $3n$ when considering limits in terms of jerk or torque time derivative. Going beyond offline sampling, one may want to learn efficient braking maps [24]. However, the dimension of the state space and the emergence of interactive contexts in robotics, where the future trajectory may not be known beforehand, render this kind of approach difficult to apply for the whole workspace. Looking at the problem more locally, the estimation of braking capabilities could be tackled online by making use of interval analysis approaches [25], [26], [27] to estimate the worst case braking capabilities on a sliding horizon of prediction. Interval approaches are well suited for handling constraints, provide formal guarantees and can be tuned in order to find the best compromise between computation time and the precision of the approximation. In addition, interval methods could also be used to predict the future motions of both the human and robot in a given time period, providing a safer estimate of the separation distance and allowing to use the robot even more efficiently. The use of interval analysis approaches is a second future direction of research related to the work in this paper.

VI. Conclusion

To improve workspace sharing between a human and a robot, more sophisticated control algorithms must be introduced to the industry. This paper proposes a simple and fast optimization based joint velocity controller which modulates the joint velocity constraints based on the robot's braking capabilities and the separation distance between the human and the robot. The controller provides optimal task tracking while respecting joint velocity constraints, ensuring that the robot is able to brake before collision with a human. Although promising, this algorithm relies on the joint capacities of the robot provided as constants by the robot manufacturer, but in fact these capacities are state dependent. Therefore, the proposed algorithm currently only uses a subset of the full capacities of the robot. The development of state dependant models of the robot's capabilities, and more optimal trajectory modulation approaches, will allow to further improve the performance and reliability of the proposed controller.

VII. Acknowledgments

This work is part of the HARRY^2 project funded by the European Union's Horizon 2020 research and innovation program within the framework of the COVR initiative under grant agreement No 779966 and the French Région Nouvelle-Aquitaine under grant agreement No 2019-1R5-0116.

REFERENCES

[1] Robotiq, “Collaborative robots buyer’s guide,” 2018. [Online]. Available: https://blog.robotiq.com/collaborative-robot-ebook

[2] A. Albu-Schäffer, ., S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, “The DLR lightweight robot: design and control concepts for robots in human environments,” *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[3] ISO/TS-10218-2, *Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration*, International Organization for Standardization, Geneva, Switzerland, 2011.

[4] ISO/TS-15066, *Robots and robotic devices - Collaborative robots*, International Organization for Standardization, Geneva, Switzerland, 2016.

[5] J. A. Marvel and R. Norcross, “Implementing speed and separation monitoring in collaborative robot workcells,” *Robotics and computer-integrated manufacturing*, pp. 144–155, 2017.

[6] F. Vicentini, M. Giussani, and L. M. Tosatti, “Trajectory-dependent safe distances in human-robot interaction,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sep. 2014, pp. 1–4.

[7] C. Vogel, C. Walter, and N. Elkmann, “A projection-based sensor system for safe physical human-robot collaboration,” in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 5359–5364.

[8] A. Cherubini, R. Passama, A. Meline, A. Crosnier, and P. Fraisse, “Multimodal control for human-robot cooperation,” in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 2202–2207.

[9] P. A. Lasota, G. F. Rossano, and J. A. Shah, “Toward safe close-proximity human-robot interaction with standard industrial robots,” in *IEEE International Conference on Automation Science and Engineering*, 2014, pp. 339–344.

[10] H. Jochen and A. Zelinsky, “Quantitative Safety Guarantees for Physical Human – Robot Interaction,” *The International Journal of Robotics Research*, vol. 22, no. 7, pp. 479–504, 2003.

[11] M. Laffranchi, N. G. Tsagarakis, and D. G. Caldwell, “Safe human robot interaction via energy regulation control,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 35–41.

[12] A. Meguenani, V. Padois, J. Da Silva, A. Hoarau, and P. Bidaud, “Energy based control for safe human-robot physical interaction,” in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 809–818.

[13] L. Joseph, V. Padois, and G. Morel, “Towards X-ray medical imaging with robots in the open: safety without compromising performances,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01614508

[14] S. Rubrecht, V. Padois, P. Bidaud, M. De Broissia, and M. Da Silva Simoes, “Motion safety and constraints compatibility for multibody robots,” *Autonomous Robots*, vol. 32, no. 3, pp. 333–349, 2012.

[15] F. Flacco, A. D. Luca, and O. Khatib, “Motion Control of Redundant Robots under Joint Constraints: Saturation in the Null Space,” in *IEEE International Conference on Robotics and Automation*, 2012.

[16] “Ros integration for franka emika research robots,” https://github.com/frankaemika/franka_ros, 2019.

[17] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, “Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, Oct 2019.

[18] “Orocos kinematics and dynamics c++ library,” https://github.com/orocos/orocos_kinematics_dynamics, 2019.

[19] “Human tracking for multi-kinect systems around a robot on ros,” https://github.com/kuka-isir/kinects_human_tracking, 2015.

[20] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, “qpOASES: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[21] A. Del Prete, “Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 281–288, 2017.

[22] A. Meguenani, “Safe control of robotic manipulators in dynamic contexts,” Ph.D. dissertation, Universite Pierre et Marie Curie, Paris 6, 2017.

[23] N. Mansfeld, M. Hamad, M. Becker, A. G. Marin, and S. Haddadin, “Safety map: A unified representation for biomechanics impact data and robot instantaneous dynamic properties,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1880–1887, 2018.

[24] O. Sigaud, C. Salaün, and V. Padois, “On-line regression algorithms for learning mechanical models of robots: a survey,” *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1115–1129, 2011.

[25] L. Jaulin, I. Braems, and E. Walter, “Interval methods for nonlinear identification and robust control,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4, Dec 2002, pp. 4676–4681 vol.4.

[26] J.-P. Merlet, “Interval Analysis and Reliability in Robotics,” *International Journal of Reliability and Safety*, vol. 3, no. 1/2/3, pp. 104–130, 2009.

[27] T. Raïssi, N. Ramdani, and Y. Candau, “Bounded error moving horizon state estimator for non-linear continuous-time systems: application to a bioprocess system,” *Journal of Process Control*, vol. 15, no. 5, pp. 537 – 545, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095915240400112X